

Structure of frequent itemsets with extended double constraints

Truong Chi Tin¹ · Duong Van Hai¹ · Hoang Nguyen Thuy Ngan¹

Received: 8 July 2015 / Accepted: 12 December 2015 / Published online: 29 January 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract Frequent itemset discovering has been one essential task in data mining. In the worst case, the cardinality of the class of all frequent itemsets is of exponent which leads to many difficulties for users. Therefore, a model of constraint-based mining is necessary when their needs and interests are the top priority. This paper aims to find a structure of frequent itemsets that satisfy the following conditions: they include a subset C_{10} , contain no items of a subset C'_{11} , and have at least an item belonging to subset C'_{21} . The first new point of the paper is the proposed theoretical result that is the generalization of our former researches (Hai et al. in Adv Comput Methods Knowl Eng Sci 479:367–378, 2013). Second, based on new sufficient and necessary conditions discovered just for closed itemsets and their generators in association with the methods of creating borders and eliminating branches and nodes on the lattice, we can effectively and quickly eliminate not only a class of frequent itemsets but also one or more branches of equivalence classes of which elements are insatiate the constraints. Third, a structure and a unique representation of frequent itemsets with extended double constraints are shown by representative closed itemsets and their generators. Finally, all theoretical results in this paper are proven to be reliable and they are firm bases to guarantee the correctness and efficiency of a new algorithm, *MFS-EDC*, which is used to effectively mine all constrained frequent itemsets. Experiments show the outstanding efficiency of this

new algorithm compared to modified post-processing algorithms on benchmark datasets.

Keywords Frequent itemset · Closed frequent itemset · Closed itemsets lattice · Generator · Double constraint · Extended double constraint · Constraint mining

1 Introduction

One of the most basic tasks in Data Mining is to discover the groups of items, products, symptoms and so on, that appear together in the given datasets. For this work, mining frequent itemset, researched first by Agrawal et al. [1] in 1993, has become more and more important and many new algorithms or improvements have been proposed to solve the problem more efficiently, such as Eclat [36], FP-Growth [18], FP-Growth* [14], BitTable-FI [12] and Index-BitTableFI [29].

A main difficulty of frequent itemset mining is that the cardinality of the solution set in the worst case is of exponent ($O(2^m)$, where $m = |A|$ and A is a set of items appearing in transactions) that can lead to the quite high computational and memory costs of mining algorithms. In fact, users can only care about a small number of them satisfying some given constraints. A model of constraint-based mining has thus been developed [5, 24]. Constraints help to focus on interesting knowledge and to reduce the number of patterns extracted to those of potential interest. In addition, they are used for decreasing the search space and enhancing the mining efficiency. There are many different kinds of constraints such as knowledge-based constraints, data constraints, dimensional constraints, interestingness constraints and rule formation constraints [23]. In relation to the properties of constraints, two important types have been studied, namely anti-monotone constraints [24], denoted as \mathcal{C}_{am} , and

✉ Duong Van Hai
haidv@dlu.edu.vn

Truong Chi Tin
tintc@dlu.edu.vn

Hoang Nguyen Thuy Ngan
thuyngan.dl7@gmail.com

¹ Department of Mathematics and Informatics,
University of Dalat, Dalat, Vietnam

monotone constraints [28], denoted as C_m . An itemset satisfies a constraint C_{am} (or C_m) if its arbitrary subset (or superset) also satisfies the constraint. C_{am} is simple and suitable with Apriori-like algorithms, so it is often integrated into them to prune candidates. On the contrary, C_m is more complicated to exploit and less effective for pruning the search space.

Most previous approaches mine frequent itemsets with either C_{am} or C_m . Mining frequent itemsets with both C_{am} and C_m is of interest because, in fact, to come closer with users' true needs, quite many various kinds are used. This can be accomplished by first mining frequent itemsets that satisfy C_{am} using algorithms, such as Apriori [1,22], Eclat [34], FP-growth [27], and then filtering the ones matching C_m in a post-processing step. This approach is inefficient because it often has to test a large number of itemsets. A more complicated solution is to integrate both C_{am} and C_m into the algorithm to find all frequent itemsets satisfying them. However, authors in [20] showed that the integration of C_m can lead to a reduction in the pruning of C_{am} since their properties are opposite. Therefore, many authors have found difficulties when facing to a quite complicated conjunction of C_{am} and C_m . An impressed approach is to combine between constraint properties and the condensed representation of frequent itemsets, such as maximal ones [11,21], closed ones or generators [7,8,26,32,33]. Instead of mining all frequent itemsets, only a small number of the condensed ones are extracted. Condensed representation has three primary advantages. First, it is easier to store because the number of condensed ones is much smaller than the size of the class of all frequent ones, especially on dense datasets. Second, we exploit it only once even when the constraints are changed. And last, the condensed representation can be used to generate all frequent ones and this generation can be performed without any access to the original dataset. In [9], the authors proposed a generic algorithm to exploit frequent itemset with both C_{am} and C_m using the minimal itemsets (like generators). They claimed that there is a tradeoff in using two of these kinds of constraints concurrently, and thus it is sometimes better to use a 'generate and test' strategy. In [10], Bucila et al. pushed both C_{am} or C_m into algorithm *DualMiner* and used the concept of positive border as a condensed representation. Unfortunately, it has to scan the dataset many times as well as perform a huge number of useless tests on long itemsets, especially when the minimum support is low. An Apriori-like algorithm, called *ExAMiner* [6], uses both of these constraints to reduce not only the input data but also the search space. However, its main difficulty is to be executed again whenever the constraints are changed. Thus, the system is hard to immediately return solution sets to users.

In this paper, we are interested in a problem that includes a conjunction of C_{am} and C_m , and each comprises of different specific constraints. Then, using closed itemsets and genera-

tors as condensed representations, we propose a new model to deal with the problem presented.

1.1 Problem statement

Before formally describing the current problem, let us present practical examples that motivate us to study and propose the new results in this paper. Let us consider searching documents on the Internet where the finding needs of users are very diverse. The datasets of information regarding documents are usually saved into the tables. Each row in a table can contain keywords, appeared in a document, the author names of the document, the type of document (Article, Book, Sort Survey, and so on) and the research area. It is common that *online users usually take interest in looking for documents that comprise of a given set of keywords. Expected documents also have to be related to a specific topic A (including one or more keywords belonging to A), but they are not involved in other topic B (not having any keywords belonging to B)*. A specific example of this problem is as follows. An online user wants to look for research results from websites or search engines such as CiteSeerX, Springer and ScienceDirect. His/her need is to find the results related to keywords in the set $C'_{21} = \{\text{'sequential patterns'}, \text{'frequent sequence'}, \text{'web usage mining'}, \text{'sequential rules'}\}$, but they are not of authors in the list $C'_{11} = \{\text{'Peter'}, \text{'Chan'}, \text{'Carmona'}, \text{'Matthews'}\}$. In addition, the found results are also 'Article' and belong to the area of 'Computer Science'. Specifically, the purpose of this user is to find articles in the field of computer science (the results have to contain both words in the set $C_{10} = \{\text{'Article'}, \text{'Computer Science'}\}$) that comprise of at least a keyword of C'_{21} and contain no authors in the list C'_{11} . It is clear that the need above is practical and the interest of many researchers nowadays. Other example for the current problem is when we want to build a filter to allow children searching interesting movies on the internet. Then, a compulsory key to obtain desired results is one in the set $C_{10} = \{\text{'movie'}\}$. Here, we only allow them to watch movies that belong to kinds in the list, $C'_{21} = \{\text{'Animated'}, \text{'Cartoons'}, \text{'Comedy'}, \text{'Fiction'}, \text{'Documentary'}\}$, but they are forbidden to see types of movies in the list $C'_{11} = \{\text{'Action'}, \text{'Horror'}, \text{'Violated'}, \text{'Sexual'}, \text{'Thriller'} \text{ and } \text{'Porn'}\}$. In other words, the aim is to allow children to find all enjoyable and good videos (that are pulling in large audience) that are 'movie' and follow one or more kinds in C'_{21} but are not any types in C'_{11} . In fact, it is able to see more different significant examples.

A formal statement of the problem in our current research is presented as follows.

Formal problem statement Let T be a dataset, \mathcal{A} be the set of all attributes or items in T . An itemset A is a non-empty subset of \mathcal{A} and a transaction in T is a set of items $t \in$

$2^{\mathcal{A}}$. The support of A , denoted as $\text{supp}(A)$, is the number of transactions that are superset of A . Given two threshold values, s_0 and s_1 , and three subsets, C_{10} , C_{11} and C_{21} , of \mathcal{A} such that $0 < s_0 \leq s_1 \leq 1$, $C_{10} \subseteq C_{11} \not\subseteq C_{21}$. A is called frequent if $\text{supp}(A) \in [s_0, s_1]$. The task is to find the class $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ of all the frequent itemsets that (1) includes a subset C_{10} , (2) contains no items of a subset $C'_{11} = \mathcal{A} \setminus C_{11}$ and (3) have at least an item belonging to subset $C'_{21} = \mathcal{A} \setminus C_{21}$. In our second example above, $C_{10} = \text{movies}$, $C'_{11} = L_1$ and $C'_{21} = L_2$. In other words, the goal is to discover all elements A of $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ that can be stated formally as below:

$$\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) \\ \stackrel{\text{def}}{=} \{L' \in \mathcal{FS} : C_{10} \subseteq L' \subseteq C_{11} \text{ and } L' \not\subseteq C_{21}\}.$$

Note that the current problem is an extension of many of our formerly considered problems. If $s_1 = 1$, \mathcal{FS} is the class of all frequent itemsets in the traditional meaning. When $s_1 < 1$, we desire to consider frequent itemsets with supports that are not too high because they sometimes are valuable. For instance, they can help discover association rules with high confidences from abnormal phenomena appearing in frequent itemsets of which frequency is not necessary to be quite high (such as new, unusual rules for both positive and negative aspects in the field of network security or for finding out the falsehood on the figure of socioeconomic field). In addition, when the constraints are given special values, we obtain frequent itemsets without any constraint or with single constraints in simple forms, $\mathcal{FS}(s_0)$ and $\mathcal{FS}_{C_{11}}(s_0)$ in [2] or $\mathcal{FS}_{\not\subseteq C_{21}}(s_0)$ in [3] or $\mathcal{FS}_{\supseteq C_{10}}(s_0)$ in [16], or with double constraints presented in $\mathcal{FS}_{C_{10} \subseteq C_{11}}(s_0, s_1)$ [17]. So, we can find that the extended double constraint presented in this paper is more general than that, $\mathcal{FS}_{C_{10} \subseteq C_{11}}(s_0, s_1)$, shown in [17], by extending a new kind of constraint set, C_{21} , which is an arbitrary subset of \mathcal{A} . Indeed, when we assign $C_{21} = \emptyset$, we immediately obtain the problem $\mathcal{FS}_{C_{10} \subseteq C_{11}}(s_0, s_1)$. The extension of the new constraint C_{21} or C'_{21} has multiple practical meanings as shown in the examples above. A quite naïve thought for solving the problem $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ is to filter from the results of $\mathcal{FS}_{C_{10} \subseteq C_{11}}(s_0, s_1)$ those that satisfy the constraint C_{21} in a post-processing step. However, this will do so many useless tests, and as a result, it will take much mining time. Thus, using the lattice of closed itemsets and their generators which is also used in [2, 3, 15–17], we study and propose a new method to effectively mine frequent itemsets with above constraints, called extended double constraints (EDC). EDC can be categorized into two kinds, $C_m(\text{supp}(L') \leq s_1, C_{10} \subseteq L' \text{ and } L' \not\subseteq C_{21})$ and $C_{am}(\text{supp}(L') \geq s_0 \text{ and } L' \subseteq C_{11})$. Below are my contributions for the method to effectively discover frequent itemsets with EDC.

1.2 Contributions

The contributions of this paper are as follows. *First*, the result of the paper is the generalization of our former problems which are to find frequent itemsets without constraints or with simpler constraints. Particularly, it is an extension of the problem $\mathcal{FS}_{C_{10} \subseteq C_{11}}(s_0, s_1)$ [17] of which the result has been published in a good international journal by further considering a significant constraint, C_{21} or C'_{21} .

Second, we showed sufficient and necessary conditions for the non-emptiness of the solution set. The conditions allow us to turn checking the constraints on the very large number of frequent itemsets into testing them on representative closed itemsets of equivalence classes with quite small amount. Our lattice-based approach becomes more effective when the conditions are combined with the techniques of creating upper and under borders to quickly eliminate branches or nodes on the lattice. It also has a high sustainability in face of regular changes of the constraints following user's need. *Third*, we show a structure and a unique representation of frequent itemsets with EDC that help us test the conditions on the quite small number and size of generators. This representation also allows us to integrate the constraints into the process of generating constrained frequent itemsets without checks in a post-processing step. *Finally*, in practice, based on these theoretical results, we propose a new algorithm, *MFS-EDC*, to completely and distinctly exploit all frequent itemsets with EDC. The advantages of *MFS-EDC* are that: it quickly discovers all frequent itemsets that satisfy opposite constraints, C_{am} and C_m , concurrently by pushing the constraints into *MFS-EDC* without direct checks on them (post-processing or naïve approaches can do this by directly testing the output results of $\mathcal{FS}_{C_{10} \subseteq C_{11}}(s_0, s_1)$ on C_{21} or the ones of $\mathcal{FS}(s_0)$ on all constraints of EDC); it is easy to be turned into parallel algorithms to obtain real time in mining process; it only needs to access the original dataset once, even if the constraints are changed regularly. This considerably enhances mining performance.

The rest of this paper is organized as follows. Some preliminary concepts related to the problem are reviewed in Sect. 2. Approaches to deal with the current problem are also considered in this section. Section 3 presents a rough partition and then a stricter partition of the solution set. In Sect. 4, we propose a structure of the solution set based on a necessary and sufficient condition of closed itemsets and their generators for the emptiness of $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$, and a unique representation of frequent itemsets with extended double constraint in each equivalence class based on closed itemsets and their generators. We also propose an efficient algorithm *MFS_EDC* to exploit all frequent ones with extended double constraint. Experimental results will be discussed in Sect. 5. The conclusions and future work is presented in Sect. 6. Finally, for easier to read, the

proof of the theoretical results in this study is moved to “Appendix”.

2 Preliminary concepts and some approaches to the problem

2.1 Preliminary concepts

For a binary dataset according to discovered data context $\mathcal{T} \stackrel{\text{def}}{=} (\mathcal{O}, \mathcal{A}, \mathcal{R})$, where \mathcal{O} is a non-empty set of transactions, \mathcal{A} is the set of all items appearing in those transactions and \mathcal{R} is a binary relation on $\mathcal{O} \times \mathcal{A}$. A set of items is called an itemset. Consider two Galois connection operators $\lambda : 2^{\mathcal{O}} \rightarrow 2^{\mathcal{A}}$ and $\rho : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{O}}$ defined as follows: $\forall \mathcal{O}, \mathcal{A} : \emptyset \neq \mathcal{O} \subseteq \mathcal{O}, \emptyset \neq \mathcal{A} \subseteq \mathcal{A}, \lambda(\mathcal{O}) \stackrel{\text{def}}{=} \{a \in \mathcal{A} | (o, a) \in \mathcal{R}, \forall o \in \mathcal{O}\}, \rho(\mathcal{A}) \stackrel{\text{def}}{=} \{o \in \mathcal{O} | (o, a) \in \mathcal{R}, \forall a \in \mathcal{A}\}$ and, as convention, $\lambda(\emptyset) \stackrel{\text{def}}{=} \mathcal{A}, \rho(\emptyset) \stackrel{\text{def}}{=} \mathcal{O}$. We denote $h(\mathcal{A}) \stackrel{\text{def}}{=} \lambda(\rho(\mathcal{A}))$ as the closure of \mathcal{A} (h is called the closure operation in $2^{\mathcal{A}}$). An itemset \mathcal{A} is called closed itemset iff¹ $h(\mathcal{A}) = \mathcal{A}$ [25]. The support of $\mathcal{A} \subseteq \mathcal{A}$, denoted as $\text{supp}(\mathcal{A})$, is the ratio of cardinality $|\rho(\mathcal{A})|$ to $|\mathcal{O}|$, i.e. $\text{supp}(\mathcal{A}) \stackrel{\text{def}}{=} |\rho(\mathcal{A})|/|\mathcal{O}|$. The minimum and maximum support thresholds are denoted as s_0 and s_1 , respectively, where $0 < 1/n \leq s_0 \leq s_1 \leq 1$ and $n \stackrel{\text{def}}{=} |\mathcal{O}|$. We only consider non-trivial items in $\mathcal{A}^{\mathcal{F}} \stackrel{\text{def}}{=} \{a \in \mathcal{A} | \text{supp}(\{a\}) \geq s_0\}$. Let \mathcal{CS} be the class of all closed itemsets together with their supports. With normal order relation “ \subseteq ” over subsets of \mathcal{A} , $\mathcal{LC} \stackrel{\text{def}}{=} (\mathcal{CS}, \subseteq)$ is the lattice of all closed itemsets organized by Hass diagram. A non-empty itemset \mathcal{A} (subset of $\mathcal{A}^{\mathcal{F}}$) is called frequent iff $s_0 \leq \text{supp}(\mathcal{A}) \leq s_1$. Note that if s_1 is equal to 1, then the traditional frequent itemset concept is obtained. Briefly, $\mathcal{FS} \stackrel{\text{def}}{=} \mathcal{FS}(s_0, s_1) \stackrel{\text{def}}{=} \{L' : \emptyset \neq L' \subseteq \mathcal{A}, s_0 \leq \text{supp}(L') \leq s_1\}$ denotes the class of all frequent itemsets and $\mathcal{FCS} \stackrel{\text{def}}{=} \mathcal{FCS}(s_0, s_1) \stackrel{\text{def}}{=} \mathcal{FS}(s_0, s_1) \cap \mathcal{CS}$ denotes the class of all frequent closed itemsets. For any two non-empty sets $G, \mathcal{A} : \emptyset \neq G \subseteq \mathcal{A} \subseteq \mathcal{A}, G$ is called a generator [25] of \mathcal{A} iff $h(G) = h(\mathcal{A})$ and $(h(G') \subset h(G), \forall G' : \emptyset \neq G' \subset G)$. Let $\mathcal{G}(\mathcal{A})$ be the class of all generators of \mathcal{A} . Since $\mathcal{G}(\mathcal{A})$ is non-empty and finite [4], $|\mathcal{G}(\mathcal{A})| = k$, all generators of \mathcal{A} are indexed: $\mathcal{G}(\mathcal{A}) = \{A_1, A_2, \dots, A_k\}$. Let $\mathcal{LCG} \stackrel{\text{def}}{=} \{\langle L, \text{supp}(L), \mathcal{G}(L) \rangle | L \in \mathcal{LC}\}$ be the lattice \mathcal{LC} of closed itemsets together with their generators and $\mathcal{LFCG}(s_0, s_1) \stackrel{\text{def}}{=} \{\langle L, \text{supp}(L), \mathcal{G}(L) \rangle \in \mathcal{LCG} | L \in \mathcal{FCS}(s_0, s_1)\}$ be the lattice of frequent ones and the generators.

To present an effective approach for the current problem, based on the closure operator h , we need an equivalence rela-

tion on the class of itemsets to partition the solution set into disjoint equivalence sub-classes.

Definition 1 ([31], *Equivalence relation* $\sim_{\mathcal{A}}$ over $\mathcal{FS}(s_0, s_1)$) Consider the following binary relation $\sim_{\mathcal{A}}$ on $\mathcal{FS}(s_0, s_1), \forall A, B \in \mathcal{FS}(s_0, s_1)$:

$$A \sim_{\mathcal{A}} B \Leftrightarrow h(A) = h(B).$$

Obviously, $\sim_{\mathcal{A}}$ is an *equivalence relation*. For each $A \in \mathcal{FS}(s_0, s_1)$, we denote $[A] \stackrel{\text{def}}{=} \{B \in \mathcal{FS}(s_0, s_1) : h(B) = h(A)\}$ as the equivalence class of all frequent itemsets having the same closure $h(A)$ and for each $L \in \mathcal{FCS}(s_0, s_1)$, we have $[L] := \{L' \subseteq L : L' \neq \emptyset, h(L') = L\}$. Using this relation, we divide $\mathcal{FS}(s_0, s_1)$ into the disjoint equivalence classes. We have the following proposition.

Proposition 1 ([31], A partition of $\mathcal{FS}(s_0, s_1)$)

$$\mathcal{FS}(s_0, s_1) = \sum_{L \in \mathcal{FCS}(s_0, s_1)} [L].$$

For each $L \in \mathcal{FCS}(s_0, s_1)$, each equivalence class $[L]$ contains frequent itemsets having the same closure L , $\rho(L)$ and especially, the same support as $\text{supp}(L)$. Moreover, this partition allows us to decrease the storage of the support of itemsets in each class, the production of duplicate candidates and the independent exploitation of each class by effective parallel algorithms in distributed environment. There are effective algorithms in the literature to mine the lattice \mathcal{LCG} such as *CHARM-L* [35] and *MinimalGenerators* [34], *Touch* [30], *GENCLOSE* [4] and *CHARM-L* and *GDP* [19].

2.1.1 Some approaches to the problem

Two post-processing approaches For the first algorithm, *MFS-PP-EDCI*, we first find the class of all frequent itemsets $A \subseteq C_{11}, A \in \mathcal{FS}_{\subseteq C_{11}}(s_0)$, by one of the well-known algorithms such as *dEclat* or *FPGrowth* with the consideration of only items belonging to C_{11} . Then, the remaining constraints, $\text{supp}(A) \leq s_1, C_{10} \subseteq A$ and $A \not\subseteq C_{21}$, are checked to generate frequent itemsets satisfying *EDC*. For the second one, *MFS-PP-EDCI*, we additionally integrate an anti-monotonic constraint, $A \not\subseteq C_{21}$, into one of the above algorithms to obtain $\mathcal{FS}_{\subseteq C_{11}, \not\subseteq C_{21}}(s_0) = \{A \in \mathcal{FS}(s_0) | A \subseteq C_{11} \text{ and } A \not\subseteq C_{21}\}$ before testing the remaining monotonic constraints, $\text{supp}(A) \leq s_1, C_{10} \subseteq A$. Note that, in fact, we often use two dualistic constraints, $A \subseteq C_{11}$ and $A \not\subseteq C_{21}$ in the form of $A \cap C'_{11} = \emptyset$ and $A \cap C'_{21} \neq \emptyset$, where $C'_{11} = \mathcal{A} \setminus C_{11}$ and $C'_{21} = \mathcal{A} \setminus C_{21}$ with the quite small sizes of C'_{11} and C'_{21} .

The drawbacks of two these approaches are taking a lot of time for mining $\mathcal{FS}_{\subseteq C_{11}}(s_0)$ or $\mathcal{FS}_{\subseteq C_{11}, \not\subseteq C_{21}}(s_0)$ again

¹ iff is denoted as if and only if.

when EDC is changed, and for direct check the remaining constraints on the large number of generated frequent itemsets. If we keep all frequent itemsets in the memory with $s_0 = 1/|\mathcal{O}|$, then that may need enormous storage, especially when $n = |\mathcal{O}|$ is quite large.

The approach of the paper Based on the partition (1) in the Proposition 2 (which divides the solution set into disjoint equivalence solution sub-classes), we first mine only once the lattice \mathcal{LCG} containing closed itemsets and their generators from \mathcal{T} . *Second*, when constraints are changed, we quickly determine from \mathcal{LCG} the class $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ of all closed frequent itemsets and generators with EDC . Note that, with this partition, instead of checking the constraints on the so large number of frequent itemsets, we just need to do that on the quite small amount of closed frequent itemsets belonging to \mathcal{LCG} . In this step, based on monotone or anti-monotone properties of constraints, the parent–child relations in the lattice are used to quickly find the supersets or the subsets of a closed itemset. That helps to significantly reduce the search space when determining the elements of $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$. Moreover, another outstanding advantage of the partition is to allow us to design parallel algorithms for concurrently, independently mining each the sub-class. *Finally*, in each equivalence sub-class $[L]$ with $L \in \mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$, we completely, quickly and distinctly generate all frequent itemsets with EDC which are represented uniquely through L and its generators $\mathcal{G}(L)$ —the essential information of the class $[L]$. From the theoretical results demonstrated to be reliable, we propose *MFS-EDC*, an efficient algorithm for mining frequent itemsets that satisfy EDC .

In next section, we first present an ineffective rough partition of the solution set, and then, based on above approach, we show a better strict partition for it.

3 Partitioning solution set by the equivalence relation

For each $L \in \mathcal{FCS}(s_0, s_1)$, $A \subseteq B \subseteq \mathcal{A}$, $C \subseteq \mathcal{A}$, $L_B \stackrel{\text{def}}{=} L \cap B$, we denote: $\mathcal{FS}_{A \subseteq L_B} \stackrel{\text{def}}{=} \{L' \in [L] \mid A \subseteq L' \subseteq L_B\} = \{L' \neq \emptyset \mid A \subseteq L' \subseteq L_B, h(L') = L\}$ and $\mathcal{FS}_{A \subseteq L_B, \not\subseteq C} \stackrel{\text{def}}{=} \{L' \in \mathcal{FS}_{A \subseteq L_B} \mid L' \not\subseteq C\}$.

3.1 The rough partition of the solution set

$$\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$$

From the partition in Proposition 1, we immediately obtain the following rough partition for $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$.

Proposition 2 (A rough partition of $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$)

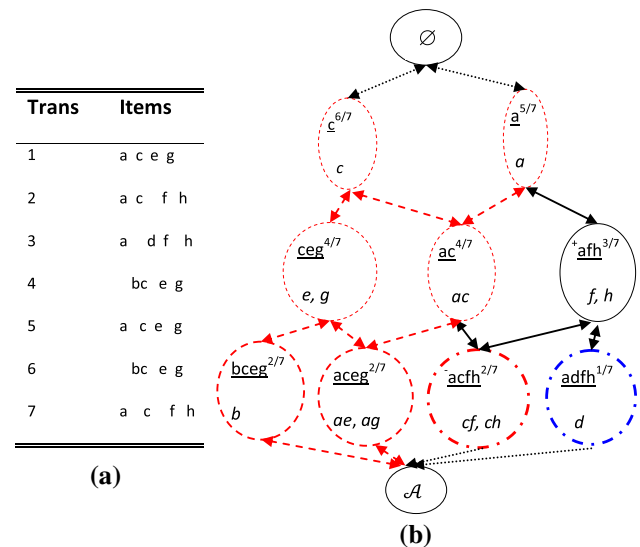


Fig. 1 a Example dataset and b the corresponding lattice of closed itemsets (underline), their generators (*italic*) and supports (^{superscript})

$$\begin{aligned} \mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) \\ = \sum_{L \in \mathcal{FCS}(s_0, s_1)} \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}. \end{aligned} \quad (1)$$

From this partition, we can independently exploit all frequent itemsets with EDC in each equivalence class $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}$.

Example 1 (Illustrating the disadvantage of the above rough partition) The rest of this paper considers dataset \mathcal{T} shown in Fig. 1a. *Charm-L* [35] and *MinimalGenerators* [34] are used to mine a lattice of all frequent itemsets and their generators.

The results are shown in Fig. 1b. Let us consider the constraints on supports $s_0 = 2/7, s_1 = 1$. For brevity, we denote itemset $\{a_1, a_2, \dots, a_k\}$ as $a_1 a_2 \dots a_k$, for example, $\{a, f, h\}$ as afh . It is possible that there are many values of constraints and closed itemsets $L \in \mathcal{FCS}(s_0, s_1)$ such that $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}} = \emptyset$, even $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) = \emptyset$. Indeed, we consider examples as follows:

- (i) For $C_{10} = a, C_{11} = adfh$ and $C_{21} = adfh$, when using the algorithm of post-processing approach based on the partition in (1), with $L = afh \in \mathcal{FCS}(s_0, s_1)$, $supp(L) = 3/7$, we have $[L] = \{f, fa, fh, fha, h, ha\}$, but $L_{C_{11}} = afh \subseteq C_{21}$, so $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}} = \emptyset$. Moreover, after generating $|\mathcal{FS}(s_0, s_1)| = 35$ itemsets and then directly testing constraints on them corresponding to all different closed frequent (CF) itemsets L in $\mathcal{FCS}(s_0, s_1)$, there is no itemset L' in any of 9 CF classes $[L]$ satisfying the constraints, so $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}} = \emptyset$ and $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}(s_0, s_1) = \emptyset$.

- (ii) We obtain the similar result $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) = \emptyset$ for other constraints, $C_{10} = d$, $C_{11} = adf$ and $C_{21} = adf$.
2. With $C_{10} = a$, $C_{11} = adfh$ and $C_{21} = ah$, for all $L \in \mathcal{FCS}(s_0, s_1)$, we only have one class $[L = afh]$ satisfying the constraints, $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}} = \{ahf, af\} \neq \emptyset$, since $\text{supp}(L) = 3/7$, $[L] = \{f, fa, fh, fha, h, ha\}$, i.e. $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) = \mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}} = \{ahf, af\} \neq \emptyset$.

Then, the post-processing approach takes quite much time to generate all frequent itemsets in equivalence sub-classes $[L]$, corresponding to closed frequent itemsets L , and then so many or even all sub-classes are eliminated because they do not satisfy the constraints. From this example, we find it is important to have sufficient and necessary conditions and then impose them on the constraints and closed itemsets L to narrow $\mathcal{FCS}(s_0, s_1)$ into $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$. Here, $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ includes closed itemsets which are not only frequent but they also satisfy constraints regarding sub-items $\{C_{10}, C_{11}, C_{21}\}$ so that corresponding solution subsets are not empty, i.e. $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}} \neq \emptyset$.

3.2 Strict partition of solution set

$$\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$$

To briefly present the remaining results, we consider the following lemma which will be used to prove Propositions 3 and 4 in the next section.

Lemma 1 Let $A, B, L', L \subseteq \mathcal{A} : L \neq \emptyset, B \neq \emptyset$,

(a)

- (i) $L' \neq \emptyset \Leftrightarrow \mathcal{G}(L') \neq \emptyset$. If $\emptyset \subset L' \subseteq L$ and $h(L') = h(L)$, then $\mathcal{G}(L') \subseteq \mathcal{G}(L)$.
- (ii) If $\mathcal{G}_B(L) \stackrel{\text{def}}{=} \{L_i \in \mathcal{G}(L) | L_i \subseteq B\} \neq \emptyset$, then $\mathcal{G}_B(L) = \mathcal{G}(L_B)$.
- (iii) Assume that $A \subseteq B$, $\forall L \in \mathcal{FCS}_{A \subseteq B}(s_0, s_1) \stackrel{\text{def}}{=} \{L \in \mathcal{FCS}(s_0, s_1) | L \supseteq A, \mathcal{G}_B(L) \neq \emptyset\}$. Then, $\forall L' \in \mathcal{FS}_{A \subseteq B}$, $\forall L_i \in \mathcal{G}(L)$, then $\mathcal{G}(L') \subseteq \mathcal{G}_B(L)$ and $(hL_i) = h(L_B) = L$.
- (iv) $\forall U, U \in 2^{\mathcal{A}} : \emptyset \subset U \subseteq V$, we have $U \cap \text{Minimal}(V) \subseteq \text{Minimal}(U)$, i.e. if $\exists M \in U \cap \text{Minimal}(V)$, then $M \in \text{Minimal}(U)$, where the set $\text{Minimal}(U)$ consists of all minimal subsets of elements of U according to the normal order relation “ \subseteq ” on subsets.

(b) Let $\text{Cond}(L_i)$ be a logic condition expression related to L_i . Consider $L' \subseteq \mathcal{A}, L' \neq \emptyset, L' \supseteq A, h(L') = L : \exists L_i \in \mathcal{G}(L')$ and assume that $\text{Cond}(L_i)$ is true. Let $U \stackrel{\text{def}}{=} \{K_k \stackrel{\text{def}}{=} L_k \setminus A | L_k \in \mathcal{G}(L'), \text{Cond}(L_k)\}$, $V \stackrel{\text{def}}{=}$

$\{K_k \stackrel{\text{def}}{=} L_k \setminus A | L_k \in \mathcal{G}(L), \text{Cond}(L_k)\}$ -be finite sets. Then, $\emptyset \subset U \subseteq V, \emptyset \subset \text{Minimal}(U) \subseteq \text{Minimal}(V)$ and we can always assume that i is the minimum index such that $K_i \in \text{Minimal}(U)$. Moreover, $L' \supseteq A + K_i$ and $K_i \in \text{Minimal}(V)$.

(c) Let $A \subseteq L_B \stackrel{\text{def}}{=} L \cap B \neq \emptyset, K_{\min}$ be a non-empty class of subsets in $L_B (\emptyset \subset K_i \subseteq L_B, \forall K_i \in K_{\min})$, $K_U^i \stackrel{\text{def}}{=} \begin{cases} \bigcup_{K_k \in K_{\min}: k \leq i} K_k, & \text{if } i \geq 1 \\ \emptyset, & \text{if } i = 0 \end{cases}$, $K_{U,i} \stackrel{\text{def}}{=} K_U^{i-1} \setminus K_i$ and $K_{-,i} \stackrel{\text{def}}{=} L_B \setminus (A + K_U^i), \forall i \geq 1$. If $L' = A + K_i + K'_i + K'_i$, where $K_i \in K_{\min}, K'_i \subseteq K_{U,i}, K'_i \subseteq K_{-,i}$, then $A \subseteq L' \subseteq L_B$. In addition, if $K_i \stackrel{\text{def}}{=} L_i \setminus A$ with $L_i \in \mathcal{G}_B(L)$, then $h(L') = h(L_B) = L$ and $L' \in \mathcal{FS}_{A \subseteq L_B}$.

Note that in general case, the reverse of the a.(iv) above is not true. However, in the present special cases, if we add some corresponding conditions, then the reverse assertion in Lemma 1b is also true.

We denote $C^+ \stackrel{\text{def}}{=} C_{11} \cap C_{21}, C_- \stackrel{\text{def}}{=} C_{10} \cap C_{21}, C^* \stackrel{\text{def}}{=} C_{11} \setminus C_{21}$ and $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) \stackrel{\text{def}}{=} \{L \in \mathcal{FCS}(s_0, s_1) | C_{10} \subseteq L_{C_{11}} \not\subseteq C_{21}, \mathcal{G}_{C_{11}}(L) \neq \emptyset\}$. It is obvious that $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) \subseteq \mathcal{FCS}(s_0, s_1), C_- \subseteq C^+$ and for every $L' \in \mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) \neq \emptyset$, then $\emptyset \subset L' \setminus C_{21} \subseteq L_{C_{11}} \setminus C_{21} \subseteq C_{11} \setminus C_{21} \subseteq C_{11}$, $\text{supp}(C_{11}) \leq \text{supp}(L') \leq s_1, \text{supp}(C_{10}) \geq \text{supp}(L') \geq s_0$. Thus, from now on, we always assume that the following hypothesis (H_1) is satisfied:

$$0 < s_0 \leq s_1 \leq 1, \quad C_{10} \subseteq C_{11} \not\subseteq C_{21}, \\ \text{supp}(C_{11}) \leq s_1, \quad \text{supp}(C_{10}) \geq s_0. \quad (H_1)$$

Proposition 3 (Necessary and sufficient conditions of $L \in \mathcal{FCS}(s_0, s_1)$ for the emptiness of $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ and $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$, a better partition of $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$) Assume that the above hypothesis (H_1) is satisfied. Then:

- (a) $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) \neq \emptyset \Leftrightarrow \mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) \neq \emptyset$.
- (b) We obtain a better partition of $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ as follows:

$$\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) \\ = \sum_{L \in \mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)} \mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}} \cdot (2)$$

The following algorithm, *MFCS-EDC*, is to extract $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ (the set of constrained frequent closed itemsets which is the output of the algorithm) from the input data which is the lattice \mathcal{LCG} of all closed item-

Fig. 2 MFCS-EDC algorithm

```

MFCSC10⊆C11,⊄C21(s0, s1) MFCS-EDC( $\mathcal{LCG}$ , s0, s1, C10, C11, C21)
1.  $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) := \emptyset$ ;  $C^* := C_{11} \setminus C_{21}$ ;
2. for each ( $\langle L, \text{supp}(L), \mathcal{G}(L) \rangle \in \mathcal{LCG}$ ) do {
3.   if ( $s_0 \leq \text{supp}(L) \leq s_1$  and  $C_{10} \subseteq L$  and  $L \cap C_{11} \not\subseteq C_{21}$ ) then {
4.      $\mathcal{G}_{C_{11}}(L) := \{L_i \in \mathcal{G}(L) \mid L_i \subseteq C_{11}\}$ ;
5.     if ( $\mathcal{G}_{C_{11}}(L) \neq \emptyset$ ) then  $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) := \mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) + \langle L, \text{supp}(L), \mathcal{G}_{C_{11}}(L) \rangle$ ;
6.   }
7. }
8. return  $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ ;

```

sets and their generators together with user constraints, s_0, s_1, C_{10}, C_{11} and C_{21} (Fig. 2).

Remark 1 Methods to eliminate nodes and branches of nodes as well as to create monotonic upper borders and anti-monotonic under borders on the lattice \mathcal{LCG} to quickly determine $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$.

When implementing the algorithm *MFCS-EDC*, to quickly determine $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ from \mathcal{LCG} , we can use the methods of eliminating nodes and branches of nodes, or creating upper and under borders while travelling the lattice. We split the constraints in $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ into the groups: *monotonic* $C_m^{\text{def}} = (\text{supp}(L) \leq s_1 \text{ and } C_{10} \subseteq L \text{ and } L \cap C_{11} \not\subseteq C_{21})$, *anti-monotonic* $C_{am}^{\text{def}} = (\text{supp}(L) \geq s_0)$ and $C_{\text{non}}^{\text{def}} = (\mathcal{G}_{C_{11}}(L) \neq \emptyset)$ (that is neither *monotonic* nor *anti-monotonic*). An itemset satisfies a constraint C_{non} if it does not satisfy both C_{am} and C_m .

To use the advantages of the properties of C_m , when travelling in the bottom-up direction of the lattice, standing at one node L , if:

- (i) The constraint C_m is not satisfied, then we immediately wipe the branch with the root node L out of the lattice (i.e. we do not need to consider all of nodes in this branch since we know for certain that C_m will not meet its requirement on them), and go to other branch.
- (ii) C_m is satisfied and C_{non} is not, then only L is eliminated from the lattice.
- (iii) Both C_m and C_{non} are satisfied, then we still keep L on the lattice.
- (iv) The conditions of C_{am} or C_{non} are not met, then L is cut out of the lattice.
- (v) Both C_{am} and C_{non} are satisfied, then we put L on the list of anti-monotonic under borders. So, it does not need to check C_{am} on all of its predecessor nodes.

Similarly, we use five steps above when going in the top-down direction of the lattice to consider C_{am} .

Note that we often pre-select one of two groups, C_{am} and C_m , which is much more likely to be not satisfied, such as the group with more constraints in the form of AND than other. We also choose the bottom-up or the top-down first so that

eliminating branches is done before borders are created. For the considering problem, it is more suitable to select C_{am} and go in the bottom-up direction of the lattice first.

Example 2 Illustrating the good effect of the sufficient and necessary conditions so that $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) = \emptyset$ or $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}} = \emptyset$ when $L \in \mathcal{FCS}(s_0, s_1) \setminus \mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ and the effectiveness of the methods of eliminating branches and nodes, and creating borders.

1. In Example 1.1(i), we just need to find one of the conditions in (H_1) which is not met: $adf h = C_{11} \not\subseteq C_{21} = adf h$, then we immediately result $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) = \emptyset$ and $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}} = \emptyset, \forall L \in \mathcal{FCS}(s_0, s_1)$. In Example 1.1(ii), since other condition in (H_1) , $1/7 = \text{supp}(C_{10}) \geq s_0 = 2/7$, is not satisfied, we lead to the similar conclusion.
2. Consider Example 1.2, the groups of constraints C_{am} and C_m for each $L \in \mathcal{LCG}$, respectively, are $\{\text{supp}(L) \geq 2/7\}$ and $\{\text{supp}(L) \leq 1, a \subseteq L, L \cap adf h \not\subseteq ah\}$. First, we go in the bottom-up direction of \mathcal{LCG} in Fig. 1b. When using the properties of monotonic constraints, the branches started at $bceg$ and $aceg$ (circled in red by dashed lines) all are eliminated since the constraint $C_m : a \not\subseteq bceg$ and $a = aceg \cap adf h \subseteq ah$ are violated (case (i)). For $L = acf h$, the constraint $C_{\text{non}} : \mathcal{G}_{C_{11}}(L) = \{L_i \in \{cf, ch\} \mid L_i \subseteq C_{11} = adf h\} = \emptyset$, is not satisfied (case (ii)), so $L \notin \mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ and we cut L out of the lattice (circled by dotted, dashed lines in red); then, we only need to consider two remaining nodes on the lattice, afh and $adf h$. The node $adf h$ (circled by dotted, dashed lines in blue) continues to be eliminated since $C_{am} : \text{supp}(adf h) = 1/7 < 2/7$ is violated (case (iv)). Second, in \mathcal{LCG} remains only one node, afh , and we travel in both the bottom-up and top-down directions started at $L = afh$ (circled by red solid lines). After that, L is put into the lists of under and upper borders since it satisfies all constraints $C_m : C_{10} = a \subseteq L_{C_{11}} = L = afh \not\subseteq C_{21} = ah, C_{am} : \text{supp}(L) = 3/7 \geq 2/7$ and $C_{\text{non}} : \mathcal{G}_{C_{11}}(L) = \mathcal{G}(L) = \{f, h\} \neq \emptyset$. Finally, we have $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) = \{L = afh\} \neq \emptyset, [L] =$

$\{f, fa, fh, fha, h, ha\}$, so $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) = \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}} = \{ahf, af\} \neq \emptyset$.

It is able to be found from this example that, for post-processing approach, we have to take a lot of time to generate all $|\mathcal{FS}(s_0, s_1)| = 35$ frequent itemsets and then check them on the constraints about sub-items. But, we only obtain two of them, ahf and af , satisfying the constraints. Meanwhile, based on the condition $L \in \mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ and the methods of eliminating branches and nodes as well as creating borders on the lattice, we quickly wiped out eight of nine equivalence sub-classes (corresponding with 29 of 35 frequent itemsets), which did not meet the requirements of the constraints, and only need to check one node.

Note that, in the final sub-class $[L = afh]$, after generating six frequent itemsets, we have to check them on the constraints in the post-processing step which can still consume time a lot. In next section, we will show the way to partition $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}$ into two disjoint solution subsets based on dividing the generators in $\mathcal{G}_{C_{11}}(L)$ into two parts according to C_{21} .

4 Structure of the solution set

$\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ and algorithm MFS-EDC

4.1 Partition and explicit structure of each equivalence class $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}$

Going on the idea of partition, for $\forall L \in \mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$, we divide each equivalence class $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}} \stackrel{\text{def}}{=} \{L' \in [L] | C_{10} \subseteq L' \subseteq L_{C_{11}}, L' \not\subseteq C_{21}\}$ into two disjoint parts based on the partition of generators in $\mathcal{G}_{C_{11}}(L)$ as follows, $\mathcal{G}_{C_{11}}(L) = \mathcal{G}_{C^+}(L) + \mathcal{G}_{C_{11}, \not\subseteq C_{21}}(L)$, where $\mathcal{G}_{C^+}(L) \stackrel{\text{def}}{=} \{L_i \in \mathcal{G}(L) : L_i \subseteq C^+\} = \{L_i \in \mathcal{G}_{C_{11}}(L) : L_i \subseteq C_{21}\}$, $\mathcal{G}_{C_{11}, \not\subseteq C_{21}}(L) \stackrel{\text{def}}{=} \{L_i \in \mathcal{G}_{C_{11}}(L) : L_i \not\subseteq C_{21}\}$.

We first number all $K_i \in K_{\min, C \subseteq L_{C^+}} \stackrel{\text{def}}{=} \text{Minimal}\{K_i \stackrel{\text{def}}{=} L_i \setminus C_-, L_i \in \mathcal{G}_{C^+}(L), 1 \leq i \leq M\}$ from 1 to M if $\mathcal{G}_{C^+}(L) \neq \emptyset$ or set $M = 0$, if $\mathcal{G}_{C^+}(L) = \emptyset$. Then, we continue to number, from $M + 1$, subsets in $K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}} \stackrel{\text{def}}{=} \text{Minimal}\{K_i \stackrel{\text{def}}{=} L_i \setminus C_{10}, L_i \in \mathcal{G}_{C_{11}, \not\subseteq C_{21}}(L), \forall i > M\}$ and denote $K_{\min, C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}} \stackrel{\text{def}}{=} K_{\min, C \subseteq L_{C^+}} \cup K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}$.

Note that if $\mathcal{G}_{C^+}(L) = \emptyset$ (like when $C^+ = \emptyset$), then we set

$$\mathcal{FS}_{C \subseteq L_{C^+}, +L_{C^*}} \stackrel{\text{def}}{=} [L]_{C \subseteq L_{C^+}, +L_{C^*}} \stackrel{\text{def}}{=} \emptyset.$$

Otherwise, if $\mathcal{G}_{C^+}(L) \neq \emptyset$, we define: $[L]_{C \subseteq L_{C^+}, +L_{C^*}} \stackrel{\text{def}}{=} \{L' \in \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}} | \exists L_i \in \mathcal{G}_{C^+}(L) : L' \supseteq L_i\}$ and $2^{[C_{10} \setminus C_{21} \subseteq L_{C^*}]} \stackrel{\text{def}}{=} \{L' \neq \emptyset | C_{10} \setminus C_{21} \subseteq L' \subseteq L_{C^*}\}$, $K_{U, C \subseteq L_{C^+}} \stackrel{\text{def}}{=} \bigcup_{K_k \in K_{\min, C \subseteq L_{C^+}}, k \leq i} K_k$, $K_{U, C \subseteq L_{C^+}, i} \stackrel{\text{def}}{=} \begin{cases} K_{U, C \subseteq L_{C^+}}^{i-1} \setminus K_i, & \text{if } i \geq 2 \\ \emptyset, & \text{if } i = 1 \end{cases}$, $K_{-, C \subseteq L_{C^+}, i} \stackrel{\text{def}}{=} L_{C^+} \setminus (C_- + K_{U, C \subseteq L_{C^+}, i}) = L_{C^+} \setminus (C_{10} \cup K_{U, C \subseteq L_{C^+}, i})$ and $\mathcal{FS}_{C \subseteq L_{C^+}}^* \stackrel{\text{def}}{=} \{L'' \stackrel{\text{def}}{=} C_- + K_i + K'_i + K_i^\sim | K_i \in K_{\min, C \subseteq L_{C^+}}, K'_i \subseteq K_{U, C \subseteq L_{C^+}, i}, K_i^\sim \subseteq K_{-, C \subseteq L_{C^+}, i} \text{ and } (K_k \not\subseteq K_i + K'_i, \forall K_k \in K_{\min, C \subseteq L_{C^+}} : 1 \leq k < i)^{(*)}, 1 \leq i \leq M\}$.

$$\mathcal{FS}_{C \subseteq L_{C^+}, +L_{C^*}}^* \stackrel{\text{def}}{=} \left\{ L' \stackrel{\text{def}}{=} L'' + L^\sim | L'' \in \mathcal{FS}_{C \subseteq L_{C^+}}^*, L^\sim \in 2^{[C_{10} \setminus C_{21} \subseteq L_{C^*}]} \right\}. \quad (3a)$$

And if $\mathcal{G}_{C_{11}, \not\subseteq C_{21}}(L) = \emptyset$ then we assign $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, C_{20} \not\subseteq G \not\subseteq C_{21}}^* \stackrel{\text{def}}{=} [L]_{C_{10} \subseteq L_{C_{11}}, C_{20} \not\subseteq G \not\subseteq C_{21}} \stackrel{\text{def}}{=} \emptyset$. On the contrary, we denote $[L]_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}} \stackrel{\text{def}}{=} \{L' \in \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}} | (\exists L_i \in \mathcal{G}_{C_{11}, \not\subseteq C_{21}}(L) : L' \supseteq L_i) \text{ and } (L_k \not\subseteq L', \forall L_k \in \mathcal{G}_{C^+}(L))\}$, $K_{U, C_{10} \subseteq L_{C_{11}}} \stackrel{\text{def}}{=} \bigcup_{K_k \in K_{\min, C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}, k \leq i} K_k$, $K_{U, C_{10} \subseteq L_{C_{11}}, i} \stackrel{\text{def}}{=} \begin{cases} K_{U, C_{10} \subseteq L_{C_{11}}}^{i-1} \setminus K_i, & \text{if } i \geq 2 \\ \emptyset, & \text{if } i = 1 \end{cases}$, $K_{-, C_{10} \subseteq L_{C_{11}}, i} \stackrel{\text{def}}{=} L_{C_{11}} \setminus (C_{10} + K_{U, C_{10} \subseteq L_{C_{11}}, i})$ and:

$$\begin{aligned} \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^* &\stackrel{\text{def}}{=} \{L' \stackrel{\text{def}}{=} C_{10} + K_i + K'_i + K_i^\sim | K_i \in K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}, \\ &K'_i \subseteq K_{U, C_{10} \subseteq L_{C_{11}}, i}, K_i^\sim \subseteq K_{-, C_{10} \subseteq L_{C_{11}}, i} \text{ and} \\ &(K_k \not\subseteq K_i + K'_i, \forall K_k \in K_{\min, C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}} : \\ &1 \leq k < i)^{(**)}, \forall i > M\}. \end{aligned} \quad (3b)$$

$$\text{Finally, we denote } \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}^* \stackrel{\text{def}}{=} \mathcal{FS}_{C \subseteq L_{C^+}, +L_{C^*}}^* + \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, rmG \not\subseteq C_{21}}^*. \quad (3c)$$

Obviously, $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}} = [L]_{C \subseteq L_{C^+}, +L_{C^*}} + [L]_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}$.

Proposition 4 (Partition, explicit structure and unique representation of $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}$) Assume that the hypothesis (H_1) is satisfied and let $L \in \mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$, then

(i) All frequent itemsets in $\mathcal{FS}_{C_{-} \subseteq L_{C^{+}}}, \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}}, \mathcal{G} \not\subseteq C_{21}$, and thus, all ones in $\mathcal{FS}_{C_{-} \subseteq L_{C^{+}}, +L_{C^{*}}}$ and $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}$ have a unique representation and are generated distinctly;

(ii)

$$\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}} = \mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}^{*}. \quad (4)$$

(i.e. each constrained sub-class $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}$ is partitioned into two disjoint subsets $\mathcal{FS}_{C_{-} \subseteq L_{C^{+}}, +L_{C^{*}}}^{*}$, $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \mathcal{G} \not\subseteq C_{21}}^{*}$).

Example 3 (Illustrating disjoint partition and structure of $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}$) According to Example 2.1, for $L = afh \in \mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}$, (s_0, s_1) , $\mathcal{G}_{C_{11}}(L) = \{f, h\}$, we will illustrate the direct generation of two constrained frequent itemsets $\{fa, fha\}$ in $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq L_{C_{21}}}$, using (4) based on the splitting of $\mathcal{G}_{C_{11}}(L) = \mathcal{G}_{C^{+}}(L) + \mathcal{G}_{C_{11}, \not\subseteq C_{21}}(L)$, where $C^{+} = L_{C^{+}} = ah$, $\mathcal{G}_{C^{+}}(L) = \{h\}$, $\mathcal{G}_{C_{11}, \not\subseteq C_{21}}(L) = \{f\}$. Besides, $C_{-} = a$, $L_{C^{*}} = \{f\}$, $K_{\min, C_{-} \subseteq L_{C^{+}}} = \{K_1 = h\}$, $K_{U, C_{-} \subseteq L_{C^{+}}, 1} = \emptyset$, $K_{-, C_{-} \subseteq L_{C^{+}}, 1} = \emptyset$, $2[C_{10} \subseteq C_{21} \subseteq L_{C^{*}}]^{*} = \{f\}$, $\mathcal{FS}_{C_{-} \subseteq L_{C^{+}}, +L_{C^{*}}}^{*} = \{L' \stackrel{\text{def}}{=} L'' + L^{\sim} = a + h + f = afh\}$; $K_{\min, C_{10} \subseteq L_{C_{11}}, \mathcal{G} \not\subseteq C_{21}} = \{K_2 = f\}$, $K_{U, C_{10} \subseteq L_{C_{11}}, 2} = \{h\}$, $K_{-, C_{10} \subseteq L_{C_{11}}, 2} = afh \setminus \{a + hf\} = \emptyset$, $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \mathcal{G} \not\subseteq C_{21}}^{*} = \{L' = a + f + K'_2 + \emptyset \mid \emptyset \subseteq K'_2 \subseteq \{h\} \text{ and } (K_1 \not\subseteq K_2 + K'_2)\} = \{af, \text{ with } K'_2 = \emptyset\} = \{af\}$, since with $K'_2 = h$, then $h = K_1 \subseteq K_2 + K'_2 = fh$. So we reject $L' = afh$ (as this itemset was belong to $\mathcal{FS}_{C_{-} \subseteq L_{C^{+}}, +L_{C^{*}}}^{*}$ already). Thus, $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}} = \mathcal{FS}_{C_{-} \subseteq L_{C^{+}}, +L_{C^{*}}}^{*} + \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \mathcal{G} \not\subseteq C_{21}}^{*} = \{afh, af\}$.

Remark 2 (An effective way to calculate subsets $K_{U,i}$, $K_{-,i}$ when finding sets in \mathcal{FS}^{*})

(a) To calculate $\mathcal{FS}_{C_{-} \subseteq L_{C^{+}}}^{*}$ more effectively, we set $K_U^i \stackrel{\text{def}}{=} K_{U, C_{-} \subseteq L_{C^{+}}}^i$, $K_{U,i} \stackrel{\text{def}}{=} K_U^{i-1} \setminus K_i$, $K_{-,i} \stackrel{\text{def}}{=} L_{C^{+}} \setminus (C_{-} + K_U^i)$ and then find that $K_{U,i} = [(K_U^{i-2} \setminus K_{i-1}) + K_{i-1}] \setminus K_i = (K_{U,i-1} + K_{i-1}) \setminus K_i$, $K_{-,i} = K_{-,i-1} \setminus K_i$, $\forall i \geq 1$ and $K_0 \stackrel{\text{def}}{=} K_{U,0} \stackrel{\text{def}}{=} \emptyset$, $K_{-,0} \stackrel{\text{def}}{=} L_{C^{+}} \setminus C_{-}$. Thus, $K_{U,i} = \begin{cases} (K_{U,i-1} + K_{i-1}) \setminus K_i, & \text{if } 1 \leq i \leq M \\ \emptyset, & \text{if } i = 0 \end{cases}$, $K_{-,i} = \begin{cases} K_{-,i-1} \setminus K_i, & \text{if } 1 \leq i \leq M \\ L_{C^{+}} \setminus C_{-}, & \text{if } i = 0 \end{cases}$ và $K_0 \stackrel{\text{def}}{=} \emptyset$.

(b) Similarly, we also have more effective recursive expressions for $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \mathcal{G} \not\subseteq C_{21}}^{*}$. If we set $K_U^i \stackrel{\text{def}}{=} K_{U, C_{10} \subseteq L_{C_{11}}}^i$, $K_{U,i} \stackrel{\text{def}}{=} K_U^{i-1} \setminus K_i$, $K_{-,i} \stackrel{\text{def}}{=} L_{C_{11}} \setminus (C_{10} + K_U^i)$, then, for $\forall i \geq M+1 \geq 2$: $K_{U,i} = (K_{U,i-1} + K_{i-1}) \setminus K_i$, $K_{-,i} =$

$\begin{cases} K_{-,i-1} \setminus K_i, & \text{if } i \geq M+2 \\ [(L_{C^{*}} \setminus C_{10}) + K_{-,M}] \setminus K_{M+1}, & \text{if } i = M+1 \end{cases}$. Indeed, for $L_{C^{*}} = L \cap C_{11} \setminus C_{21}$, since $K_{-,M+1} = L \cap C_{11} \setminus (C_{10} + K_U^{M+1}) = [L_{C^{*}} \setminus (C_{10} + K_U^M)] \setminus K_{M+1} + [L \cap C_{11} \cap C_{21} \setminus (C_{10} + K_U^M)] \setminus K_{M+1} = [L_{C^{*}} \setminus C_{10} \setminus K_{M+1}] + [L \cap C_{11} \cap C_{21} \setminus (C_{10} \cap C_{21} + K_U^M \setminus K_{M+1})] = [L_{C^{*}} \setminus C_{10} \setminus K_{M+1}] + [L_{C^{+}} \setminus (C_{-} + K_U^M) \setminus K_{M+1}] = [(L_{C^{*}} \setminus C_{10}) + K_{-,M}] \setminus K_{M+1}$. Especially, if $M = 0$, then $K_{-,1} = [(L_{C^{*}} \setminus C_{10}) + K_{-,0}] \setminus K_1 = [(L_{C^{*}} \setminus C_{10}) + (L_{C^{+}} \setminus C_{-})] \setminus K_1 = [(L_{C^{*}} \setminus C_{10}) + (L_{C^{+}} \setminus C_{10})] \setminus K_1 = L_{C_{11}} \setminus (C_{10} + K_1)$.

* In brief, for $\forall M \geq 0$ (If $\mathcal{G}_{C^{+}}(L) \neq \emptyset$, then $M > 0$; else $M = 0$), if we set

$$K_{\min} \stackrel{\text{def}}{=} \begin{cases} K_{\min,1} = K_{\min, C_{-} \subseteq L_{C^{+}}}, & \text{if } \text{Begin} = 0 \\ K_{\min,2} = K_{\min, C_{10} \subseteq L_{C_{11}}, \mathcal{G} \not\subseteq C_{21}}, & \text{if } \text{Begin} = M \end{cases}$$

$$K_{\min}^{+} \stackrel{\text{def}}{=} K_{\min,1} + K_{\min,2},$$

$$K_{\min-}^{+} \stackrel{\text{def}}{=} \begin{cases} K_{\min,1}, & \text{if } \text{Begin} = 0 \\ K_{\min}^{+}, & \text{if } \text{Begin} = M \end{cases};$$

$$K_{U,i} = \begin{cases} (K_{U,i-1} + K_{i-1}) \setminus K_i, & \text{if } i \geq 2 \\ K_{U,1}, & \text{if } i = 1 \end{cases}, \forall \text{Begin} \in \{0; M\} \text{ and } K_{U,0} = \emptyset;$$

$$K_{-,i} = \begin{cases} K_{-,i-1} \setminus K_i, & \text{if } i \geq \text{Begin} + 2 \\ \text{First}_{-} \setminus K_i, & \text{if } i = \text{Begin} + 1 \end{cases}, \forall \text{Begin} \geq 0$$

$$\text{where } \text{First}_{-} = \begin{cases} L_{C^{+}} \setminus C_{-}, & \text{if } \text{Begin} = 0 \\ (L_{C^{*}} \setminus C_{10}) + K_{-,M}, & \text{if } \text{Begin} = M \end{cases}$$

and

$$K_{-,0} \stackrel{\text{def}}{=} L_{C^{+}} \setminus C_{-};$$

$$\text{Sub}_{-}L = \begin{cases} C_{-}, & \text{if } \text{Begin} = 0 \\ C_{10}, & \text{if } \text{Begin} = M \end{cases}.$$

Then, we can calculate L' in $\mathcal{FS}_{C_{-} \subseteq L_{C^{+}}}^{*}$, and L' in $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \mathcal{G} \not\subseteq C_{21}}^{*}$ as follows

$L' \stackrel{\text{def}}{=} \text{Sub}_{-}L + K_i + K'_i + K_i^{\sim}$, where $K_i \in K_{\min,M}$, $K'_i \subseteq K_{U,i}$, $K_i^{\sim} \subseteq K_{-,i}$ and $(K_k \not\subseteq K_i + K'_i, \forall K_k \in K_{\min}, C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21} : 1 \leq k < i)$, $\forall i \geq \text{Begin} + 1$, using the general algorithm below.

$$\mathcal{FS}_{\text{Sub}_{-}L}^{*} = \text{MFS} - \text{EDC} - \text{SubClass}$$

$$(\text{Sub}_{-}L, K_{\min}, \&\text{Begin}, \&K_{U,\text{Begin}}, \&K_{-, \text{Begin}}, K_{\min-}^{+}).$$

Then

$$\mathcal{FS}_{C_{-} \subseteq L_{C^{+}}}^{*} = \text{MFS-EDC-SubClass}$$

$$(C_{-}, K_{\min,1}, \text{Begin} = 1, K_{U,\text{Begin}}, K_{-, \text{Begin}}, K_{\min,1})$$

And after the algorithm finishes, we obtain $\text{Begin} = M$, $K_{U,\text{Begin}} = K_{U,M}$, $K_{-, \text{Begin}} = K_{-,M}$. After that, we calculate new values, $\text{Begin} = M + 1$, $K_{U,\text{Begin}} = (K_{U,M} + K_M) \setminus K_{M+1}$, $K_{-, \text{Begin}} = K_{-,M} \setminus K_{M+1}$, to find $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \mathcal{G} \not\subseteq C_{21}}^{*} = \text{MFS-EDC-SubClass}(C_{10}, K_{\min,2}, \text{Begin} = M + 1, K_{U,\text{Begin}}, K_{-, \text{Begin}}, K_{\min-}^{+})$.

(c) Note that, the condition (**) is not satisfied if $\exists K_i \in K_{\min, C_{10} \subseteq L_{C_{11}}, \mathcal{G} \not\subseteq C_{21}} \cap K_{\min, C_{-} \subseteq L_{C^{+}}}$. Thus,

```

FSSub-L MFS-EDC-SubClass(Sub-L, Kmin, &KBeg, &KUBeg, &K_Beg, Kmin+)
1. FSSub-L := ∅;
2. if (Kmin) then return ∅;
3. KU,Beg := KUBeg; K-Beg := K_Beg; // remark 2.b
4. for (i=Beg+1; Ki ∈ Kmin+; i++) do {
5.   KU,i := (KU,i-1 + Ki-1) ∪ Ki; // remark 2.b
6.   for each (Ki ⊆ KU,i) do {
7.     IsDuplicate := false; // saving only once A ⊆ Ki + Ki+
8.     for (j=1; Kj ∈ Kmin+; j < i; j++) do
9.       if (Kj ⊆ Ki + Ki+) then {
10.        IsDuplicate := true; break;
11.      }
12.     if (not(IsDuplicate)) then {
13.       K-i := K-i-1 ∪ Ki; // remark 2.b
14.       for each (Ki ⊆ K-i) do
15.         FSSub-L := FSSub-L ∪ {Sub-L + Ki + Ki+ + Ki+};
16.       }
17.     } // for each Ki+
18.   } // for i
19. Beg = i-1; KUBeg := KU,Beg; K_Beg := K-Beg;
20. return FSSub-L;

```

Fig. 3 MFS-EDC-SubClass algorithm

$K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}$ and $K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^+$ can be replaced by $K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}$ $\stackrel{\text{def}}{=} \text{Minimal}\{K_i \stackrel{\text{def}}{=} L_i \setminus C_{10}, L_i \in \mathcal{G}_{C_{11}, G \not\subseteq C_{21}}(L), \forall i > M\} \setminus K_{\min, C_- \subseteq L_{C^+}}$ and $K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^+ \stackrel{\text{def}}{=} K_{\min, C_- \subseteq L_{C^+}} + K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}$, respectively (Fig. 3).

(d) Since K_k and K_i are two minimal subsets belonging to different sets $K_{\min, C_- \subseteq L_{C^+}}$ and $K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}$, respectively, so they can still get equal values and we are unable to replace the sign $\not\subseteq$ in (**) by \subseteq , i.e. if $K_k = K_i' + K_i^{\sim}$, then we still eliminate K_i' . Indeed, consider the example in [4, 16], for $L = acdfh$, $\mathcal{G}(L) = \{cd, acf\}$, $\text{supp}(L) = 1/6$, $C_{10} = afd$, $C_{11} = acdf$, $C_{21} = cd$, $s_0 = 1/6$, $s_1 = 1$. Then, we have $C^+ = cd$, $C_- = d$, $L_{C^*} = af$, $\mathcal{G}_{C_{11}}(L) = \mathcal{G}(L)$, $\mathcal{G}_{C^+}(L) = \{L_1 = cd\}$, $\mathcal{G}_{C_{11}, G \not\subseteq C_{21}}(L) = \{L_2 = acf\}$, $K_{\min, C_- \subseteq L_{C^+}} = \{K_1 = c\}$, $K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}} = \{K_2 = c\}$, $\emptyset \subset af \subseteq L^{\sim} \subseteq af$, $K_{U, C_- \subseteq L_{C^+}, 1} = \emptyset$, $K_{-, C_- \subseteq L_{C^+}, 1} = acdf \setminus (afd + c) = \emptyset$, $\mathcal{FS}_{C_- \subseteq L_{C^+}, +L_{C^*}}^* = \{L'' + L^{\sim} = (d + c + \emptyset + \emptyset) + af\} = \{dcdf\}$; $K_2' \subseteq K_{U, C_{10} \subseteq L_{C_{11}}, 2} = K_1 \setminus K_2 = \emptyset$, $K_2^{\sim} \subseteq K_{-, C_{10} \subseteq L_{C_{11}}, 2} = acdf \setminus (afd + c) = \emptyset$, $K_2' = \emptyset$, $K_2^{\sim} = \emptyset$, $K_1 = K_2 + \emptyset = c$ and then obtain a frequent itemset $L' = afd + c = afdc$ which coincides with $L' = dcdf$ in $\mathcal{FS}_{C_- \subseteq L_{C^+}, +L_{C^*}}^*$. Therefore, we still wipe out L' corresponding with $K_2' = \emptyset$ and $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^* = \emptyset$. Hence, $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^* = \mathcal{FS}_{C_- \subseteq L_{C^+}, +L_{C^*}}^* = \{dcdf\}$.

Theorem 1 (Structure of the solution set $\mathcal{FS}_{C_{10} \subseteq C_{11}, G \not\subseteq C_{21}}(s_0, s_1)$)

Assume that the hypothesis (H₁) is satisfied. Then

$$\mathcal{FS}_{C_{10} \subseteq C_{11}, G \not\subseteq C_{21}}(s_0, s_1) = \sum_{L \in \mathcal{FCS}_{C_{10} \subseteq C_{11}, G \not\subseteq C_{21}}(s_0, s_1)} \times \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^* \quad (5)$$

Proof It is a consequence of (2) in Proposition 3 and (4) in Proposition 4. \square

Remark 3 (Some typically special cases) When the constraints are gotten special values, we obtain better results than those known in our former papers (since the way of calculating sets K_U^i , $K_{U,i}$ and $K_{-,i}$ to find solution sub-classes in this new version will take time less and thus more effective than the former one, according to Remark 2).

(i) To show the structure of frequent itemsets with simple double constraint $\mathcal{FS}_{C_{10} \subseteq C_{11}}(s_0, s_1)$ in [17], we choose $C_{21} = \emptyset$, with conditions changed as follows: $C^* = C_{11} \neq \emptyset$, $C^+ = C_- = \emptyset$, $\mathcal{FCS}_{C_{10} \subseteq C_{11}}(s_0, s_1) \stackrel{\text{def}}{=} \{L \in \mathcal{FCS}(s_0, s_1) | C_{10} \subseteq L_{C_{11}}, L_{C_{11}} \neq \emptyset, \mathcal{G}_{C_{11}}(L) \neq \emptyset\}$. Since $\mathcal{G}_{C^+}(L) = \emptyset$, $K_{\min, C_- \subseteq L_{C^+}} = \emptyset$, $M = 0$, so $\mathcal{G}_{C_{11}}(L) = \mathcal{G}_{C_{11}, G \not\subseteq \emptyset}(L)$, $\mathcal{FS}_{C_- \subseteq L_{C^+}, +L_{C^*}}^* = \emptyset$, let

$$K_{\min, C_{10} \subseteq L_{C_{11}}} \stackrel{\text{def}}{=} K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq \emptyset}^+ = K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq \emptyset} \\ = \text{Minimal}\{K_i \stackrel{\text{def}}{=} L_i \setminus C_{10}, L_i \in \mathcal{G}_{C_{11}}(L), i \geq 1\} \\ \text{and } \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}}^* \stackrel{\text{def}}{=} \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq \emptyset}^* = \{L' \stackrel{\text{def}}{=} C_{10} + K_i + K_i' + K_i^{\sim} | K_i \in K_{\min, C_{10} \subseteq L_{C_{11}}}, K_i' \subseteq K_{U,i}, K_i^{\sim} \subseteq K_{-,i} \text{ and } (K_k \not\subseteq K_i + K_i', \forall K_k \in K_{\min, C_{10} \subseteq L_{C_{11}}} : 1 \leq k < i)\}, \text{ v\o oi} \\ K_{U,i} = \begin{cases} (K_{U,i-1} + K_{i-1}) \setminus K_i, & \text{if } i \geq 1 \\ \emptyset, & \text{if } i = 0 \end{cases}, K_{-,i} = \begin{cases} K_{-,i-1} \setminus K_i, & \text{if } i \geq 1 \\ L_{C_{11}} \setminus C_{10}, & \text{if } i = 0 \end{cases} \text{ and } K_0 \stackrel{\text{def}}{=} \emptyset \text{ (see Remark 2)}.$$

(ii) To obtain the results presented in [16] or in [2] with single constraints $\mathcal{FS}_{C_{10} \subseteq (s_0, s_1)}$ or $\mathcal{FS}_{C_{11}}(s_0, s_1)$, respectively, we only need to not consider ones C_{11} or C_{10} in (i) which can be performed by assigning $C_{11} = \mathcal{A}$ or $C_{10} = \emptyset$, respectively.

(iii) When $C_{10} = C_{21} = \emptyset$, $C_{11} = \mathcal{A}$, $s_1 = 1$, we have the structure of frequent itemsets without constraints, $\mathcal{FS}^*(s_0) = \mathcal{FS}_{\emptyset \subseteq \mathcal{A}, G \not\subseteq \emptyset}^*(s_0, s_1)$. Then, $\mathcal{G}_{C_{11}}(L) = \mathcal{G}(L)$, $\forall L \in \mathcal{FCS}_{\emptyset \subseteq \mathcal{A}, G \not\subseteq \emptyset}(s_0, s_1) = \mathcal{FCS}(s_0) = \{L \in \mathcal{CS} : \emptyset \neq L, \text{supp}(L) \geq s_0\}$ and $\mathcal{FS}_{\subseteq L}^* \stackrel{\text{def}}{=} \{L' \stackrel{\text{def}}{=} L_i + L_i' + L_i^{\sim} | L_i \in \mathcal{G}(L), L_i' \subseteq L_{U,i}, L_i^{\sim} \subseteq L_{-,i} \text{ and } (L_k \not\subseteq L_i + L_i', \forall L_k \in \mathcal{G}(L) : 1 \leq k < i)\}$, where $K_{U,i} = \begin{cases} (K_{U,i-1} + K_{i-1}) \setminus K_i, & \text{if } i \geq 1 \\ \emptyset, & \text{if } i = 0 \end{cases}$,

$$K_{-,i} = \begin{cases} K_{-,i-1} \setminus K_i, & \text{if } i \geq 1 \\ L, & \text{if } i = 0 \end{cases} \text{ and } K_0 \stackrel{\text{def}}{=} \emptyset.$$

(iv) To show the structure of frequent itemsets with the dualistic constraint $\mathcal{FS}_{G \not\subseteq C_{21}}(s_0, s_1) \stackrel{\text{def}}{=} \{A \in \mathcal{FS}(s_0, s_1) | A \not\subseteq C_{21}\}$ in [3], we choose $C_{10} = \emptyset$, $C_{11} = \mathcal{A}$. Conditions are changed as follows: $C^* = \mathcal{A} \setminus C_{21} \neq \emptyset$, $\text{supp}(\mathcal{A}) \leq s_1$, $C^+ = C_{21}$, $C_- = \emptyset$, $\mathcal{FCS}_{C_{10} \subseteq C_{11}}(s_0, s_1) \stackrel{\text{def}}{=} \{L \in \mathcal{FCS}(s_0, s_1) | L \not\subseteq C_{21}\}$, $\mathcal{G}(L) = \mathcal{G}_{C_{21}}(L) + \mathcal{G}_{G \not\subseteq C_{21}}(L)$,

Fig. 4 MFS-EDC algorithm

```

 $\mathcal{FS}_{C_{10} \subseteq C_{11}, \notin C_{21}}(s_0, s_1)$  MFS-EDC ( $\mathcal{L}\mathcal{C}\mathcal{G}$ ,  $s_0$ ,  $s_1$ ,  $C_{10}$ ,  $C_{11}$ ,  $C_{21}$ )
1.  $\mathcal{FS}_{C_{10} \subseteq C_{11}, \notin C_{21}}(s_0, s_1) := \emptyset$ ;  $C^* := C_{11} \setminus C_{21}$ ;
2. if ( $s_0 > s_1$  or  $\text{supp}(C_{10}) < s_0$  or  $\text{supp}(C_{11}) > s_1$  or  $C_{10} \setminus C_{11} \neq \emptyset$  or  $C^* = \emptyset$ ) then return  $\emptyset$ ; // ( $H_1$ ) is not true
3.  $\mathcal{FC}_{C_{10} \subseteq C_{11}, \notin C_{21}}(s_0, s_1) := \text{MFCS\_EDC}(\mathcal{L}\mathcal{C}\mathcal{G}, s_0, s_1, C_{10}, C_{11}, C_{21})$ ;
4. for each ( $\langle L, \text{supp}(L) \rangle \in \mathcal{FC}_{C_{10} \subseteq C_{11}, \notin C_{21}}(s_0, s_1)$ ) do {
5.    $L_{C_{11}} := L \cap C_{11}$ ;
6.    $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^* := \text{MFS\_EDC\_OneClass}(C_{10}, C_{11}, C_{21}, \mathcal{G}_{C_{11}}(L))$ ;
7.    $\mathcal{FS}_{C_{10} \subseteq C_{11}, \notin C_{21}}(s_0, s_1) := \mathcal{FS}_{C_{10} \subseteq C_{11}, \notin C_{21}}(s_0, s_1) + \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^*$ ;
8. }
9. return  $\mathcal{FS}_{C_{10} \subseteq C_{11}, \notin C_{21}}(s_0, s_1)$ ;

```

Fig. 5 a MFS-EDC-OneClass algorithm. b MFS-EDC-FirstSubClass algorithm. c MFS-EDC-SecondSubClass algorithm

```

 $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^*$  MFS-EDC-OneClass ( $C_{10}$ ,  $C_{11}$ ,  $C_{21}$ ,  $L$ ,  $\mathcal{G}_{C_{11}}(L)$ ):
1.  $\mathcal{G}_{C^+}(L) = \{L_i \in \mathcal{G}_{C_{11}}(L) \mid L_i \subseteq C_{21}\}$ ;  $\mathcal{G}_{C_{11}, \notin C_{21}}(L) := \{L_i \in \mathcal{G}_{C_{11}}(L) \mid L_i \not\subseteq C_{21}\}$ ;
2.  $K_{min,1} = \text{Minimal} \{K_i \stackrel{\text{def}}{=} L_i \setminus C_{-}, L_i \in \mathcal{G}_{C^+}(L), 1 \leq i \leq M\}$ ; // if  $\mathcal{G}_{C^+}(L)$  khac  $= \emptyset$ , then  $M \geq 1$ , else  $M = 0$ 
3.  $K_{min,2} = \text{Minimal} \{K_i \stackrel{\text{def}}{=} L_i \setminus C_{10}, L_i \in \mathcal{G}_{C_{11}, \notin C_{21}}(L), i \geq M+1\} \setminus K_{min,1} \setminus C_{-}$ ; // Remark 2.c
4.  $K_{min}^+ = K_{min,1} + K_{min,2}$ ;
5.  $\mathcal{FS}_{C_{-} \subseteq L_{C^+}, +L_{C^*}} := \text{MFS\_EDC\_FirstSubClass}(C_{10}, C_{11}, C_{21}, L, K_{min,1}, \text{Beg}, \text{KUBeg}, K\_Beg)$ ;
6.  $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^* := \text{MFS\_EDC\_SecondSubClass}(C_{10}, C_{11}, C_{21}, L, K_{min,2}, \text{Beg}, \text{KUBeg}, K\_Beg, K_{min}^+)$ ;
7.  $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^* := \mathcal{FS}_{C_{-} \subseteq L_{C^+}, +L_{C^*}} + \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^*$ ;
8. return  $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^*$ ;

```

(a)

```

 $\mathcal{FS}_{C_{-} \subseteq L_{C^+}, +L_{C^*}}^*$  MFS\_EDC\_FirstSubClass ( $C_{10}$ ,  $C_{11}$ ,  $C_{21}$ ,  $L$ ,  $K_{min,1}$ , &Beg, &KUBeg, &K\_Beg)
1.  $\mathcal{FS}_{C_{-} \subseteq L_{C^+}, +L_{C^*}}^* := \emptyset$ ;  $K_0 = \emptyset$ ;  $K_{-0} = L_{C^+} \setminus C_{-}$ ; Beg=0; KUBeg= $\emptyset$ ; K\_Beg= $K_{-0}$ ;
2.  $L_{C^*} := (L \cap C_{11}) \setminus C_{21}$ ;  $C_{+} := C_{10} \setminus C_{21}$ ;  $C_{-} := C_{10} \cap C_{21}$ ;  $C^+ := C_{11} \cap C_{21}$ ;  $L_{C^+} = L \cap C^+$ ;
3. If ( $K_{min,1} \neq \emptyset$ ) then {
4.   Beg=1; First= $L_{C^+} \setminus C_{-}$ ; K\_Beg=First  $\setminus K_{Beg}$ ;
5.    $\mathcal{FS}_{C_{-} \subseteq L_{C^+}}^* = \text{MFS\_EDC\_SubClass}(C_{-}, K_{min,1}, \text{Beg}, \text{KUBeg}, K\_Beg, K_{min,1})$ ;
6.   for each ( $L'' \in \mathcal{FS}_{C_{-} \subseteq L_{C^+}}^*$ ) do
7.     for each ( $L' \subseteq L_{C^*} \mid C_{+} \subseteq L' \text{ and } L' \neq \emptyset$ ) do
8.        $\mathcal{FS}_{C_{-} \subseteq L_{C^+}, +L_{C^*}}^* := \mathcal{FS}_{C_{-} \subseteq L_{C^+}, +L_{C^*}}^* + \{L' + L''\}$ ;
9. }
10. return  $\mathcal{FS}_{C_{-} \subseteq L_{C^+}, +L_{C^*}}^*$ ;

```

(b)

```

 $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^*$  MFS\_EDC\_SecondSubClass ( $C_{10}$ ,  $C_{11}$ ,  $C_{21}$ ,  $L$ ,  $K_{min,2}$ , Beg, KUBeg, K\_Beg,  $K_{min}^+$ )
1. First =  $((L \cap C_{11}) \setminus C_{21} \setminus C_{10}) + K\_Beg$ ; Beg=Beg+1;
2. K\_Beg=First  $\setminus K_{Beg}$ ; KUBeg =  $(\text{KUBeg} + K_{Beg-1}) \setminus K_{Beg}$ ;
3.  $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^* = \text{MFS\_EDC\_SubClass}(C_{10}, K_{min,2}, \text{Beg}, \text{KUBeg}, K\_Beg, K_{min}^+)$ ;
4. return  $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^*$ ;

```

(c)

$$\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^* \stackrel{\text{def}}{=} \{L' \stackrel{\text{def}}{=} L'' + L' \mid L'' \in \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^*, \emptyset \subset L' \subseteq L \setminus C_{21}\}, \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^* \stackrel{\text{def}}{=} \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^* \setminus \emptyset \subseteq L, \notin C_{21}$$

$$\text{and } \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^* = \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^* + \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^*.$$

According to Propositions 3 and 4, we obtain two procedures *MFS-EDC-FirstSubClass* and *MFS-EDC-SecondSubClass* (pseudo code shown in Fig. 5a, b) to

produce constrained frequent itemsets in two sub-classes $\mathcal{FS}_{C_{-} \subseteq L_{C^+}, +L_{C^*}}^*$ and $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \notin C_{21}}^*$, respectively, and then the procedure *MFS-EDC-OneClass* (see in Fig. 5) to generate all constrained frequent itemsets in an equivalence class. Using Theorem 1 and these procedures, the algorithm *MFS-EDC* is proposed, shown in Fig. 4, for mining all frequent itemsets with *EDC*.

5 Experiments

Experiments were performed on a PC with an i5-2400 CPU, 3.10 GHz@ 3.09 GHz PC and 3.16 GB of memory, running on Windows XP. The algorithms were coded in C#. To compare the performance, the source code for *Charm-L* [35], *MinimalGenerators* [34] and *dEclat* [36] was converted to C#. *Charm-L* and *MinimalGenerators* were used to mine the lattice of the closed itemsets and their generators. *dEclat* was used to exploit all frequent itemsets.

To test and evaluate our new proposed algorithm, **MFS-EDC**, we compare its performance to those of two different new post-processing algorithms. The first one is called *MFS-E-EDC* that is a new modified version of *dEclat* for mining frequent itemsets with the extended double constraint. *MFS-E-EDC* is done by integrating constraints s_0 and C_{11} into *dEclat* algorithm to discover only frequent itemsets satisfying two these constraints. Then, *MFS-E-EDC* implements a post-processing step to filter frequent itemsets satisfying the remaining constraints, s_1 , C_{10} and C_{21} . The second new post-processing algorithm is named *MFS-PP-EDC* that is a modification of *Gen_Itemsets* [2]. *MFS-PP-EDC* includes two steps. In the first step, it uses *Gen_Itemsets* to mine all frequent itemsets without constraints. The second one is to directly check all generated frequent itemsets on the constraints to filter frequent itemsets satisfying extended double constraint.

We chose benchmark datasets in FIMDR [13] including Pumsb, Connect, Mushroom, Chess, and T40I10D100K to test the algorithms in performance. Pumsb, Connect, Chess, and Mushroom are real and dense, i.e. they produce many long frequent itemsets even for very high support values. The other is synthetic and sparse. Table 1 shows their characteristics.

We keep the support threshold s_1 unchanged at 0.9. Assuming that the size of C_{10} is m , then C_{11} with the size of $m + d \cdot |A^F| / 100$ ($d \in [1, 100]$) is chosen. For each pair of datasets (DB) and minimum support (MS), m ranges from 10 to 28 % of $|A^F|$ (step 2 %) and $d = 60$. For each pair of C_{10} 's size and C_{11} 's size, there are 10 value triples of C_{10} , C_{11} and C_{21} randomly selected from A^F (the size of C_{21} is also chosen randomly).

Table 1 Dataset characteristics

Dataset	#Items	#Records	Avg. length
Connect (C)	129	67,557	43
Mushroom (M)	119	8124	23
Pumsb (P)	7117	49,046	74
Chess (Ch)	75	3196	37
T40I10D100K (T40)	1000	100,000	40

Table 2 Time reductions of *MFS-EDC* compared to *MFS-PP-EDC* and *MFS-E-EDC*

DS-MS	R_PP (%)	R_E (%)	DS-MS	R_PP (%)	R_E (%)
M-14	1.50	6.32	Ch-76	33.06	17.58
M-12	1.44	6.86	Ch-72	25.73	27.86
M-10	0.60	9.82	C-82	4.48	5.12
M-8	0.53	10.40	C-80	23.62	6.71
M-6	0.34	11.97	C-78	13.01	5.73
P-82	65.87	5.89	C-76	9.11	6.20
P-78	26.06	9.62	C-72	4.03	6.27
P-76	17.70	12.80	T40-1	45.27	5.41
P-74	11.71	16.17	T40-0.8	42.23	5.12
P-72	8.57	23.24	T40-0.6	41.12	4.87
Ch-84	3.51	2.94	T40-0.4	39.23	4.13
Ch-82	7.54	5.82	T40-0.2	38.01	3.98
Ch-80	2.85	2.90			

Let T_{EDC} , T_{PP_EDC} , and T_{E_EDC} be the average execution times of *MFS-EDC*, *MFS-PP-EDC*, and *MFS-E-EDC* for 100 selected extended double constraints.

Table 2 shows the experimental evaluation of *MFS-EDC* against *MFS-PP-EDC* and *MFS-E-EDC*, where column *DS-MS* denotes the dataset *DS* with the minimum support *MS* (for example, $M - 14$ means the dataset Mushroom with the minimum support of 14 %), column *R_PP* shows the ratios of T_{EDC} and T_{PP_EDC} , and column *R_E* reveals the rates of T_{EDC} and T_{E_EDC} . Compared to *MFS-PP-EDC*, *MFS-EDC* is faster for all selected datasets. The time is reduced by 65.87–0.34 %. *MFS-EDC* is also much faster than *MFS-E-EDC* for all datasets with the time reduction from 27.86 to 2.90 %.

We found that the reason for the reduction in the mining time of *MFS-EDC* in comparison with *MFS-PP-EDC* and *MFS-E-EDC* is because there are a large number of candidates which fail the last test of both *MFS-PP-EDC* and *MFS-E-EDC*, leading to their lower performance. Note that, for sparse dataset T10, the time reduction of *MFS-EDC*, compared to *MFS-PP-EDC*, in general, is not high (over 54.73 %) because the number of frequent itemsets is small and their size is small too, leading to a low cost for testing the constraints. However, when compared to *MFS-E-EDC* for this dataset, the figure is quite high, accounting for over 94.59 %. This can be explained that when constraints are changed, *MFS-E-EDC* have to re-scan the original dataset, which will take a lot of mining time, while *MFS-EDC* only needs to travel back to the lattice.

Figure 6a, b show the comparisons of the average execution times for various support values. The performance and scalability of *MFS-EDC* are superior to those of *MFS-PP-EDC* and *MFS-E-EDC*.

Fig. 6 **a** Performance results for Chess and Mushroom. **b** Performance results for Pumsb

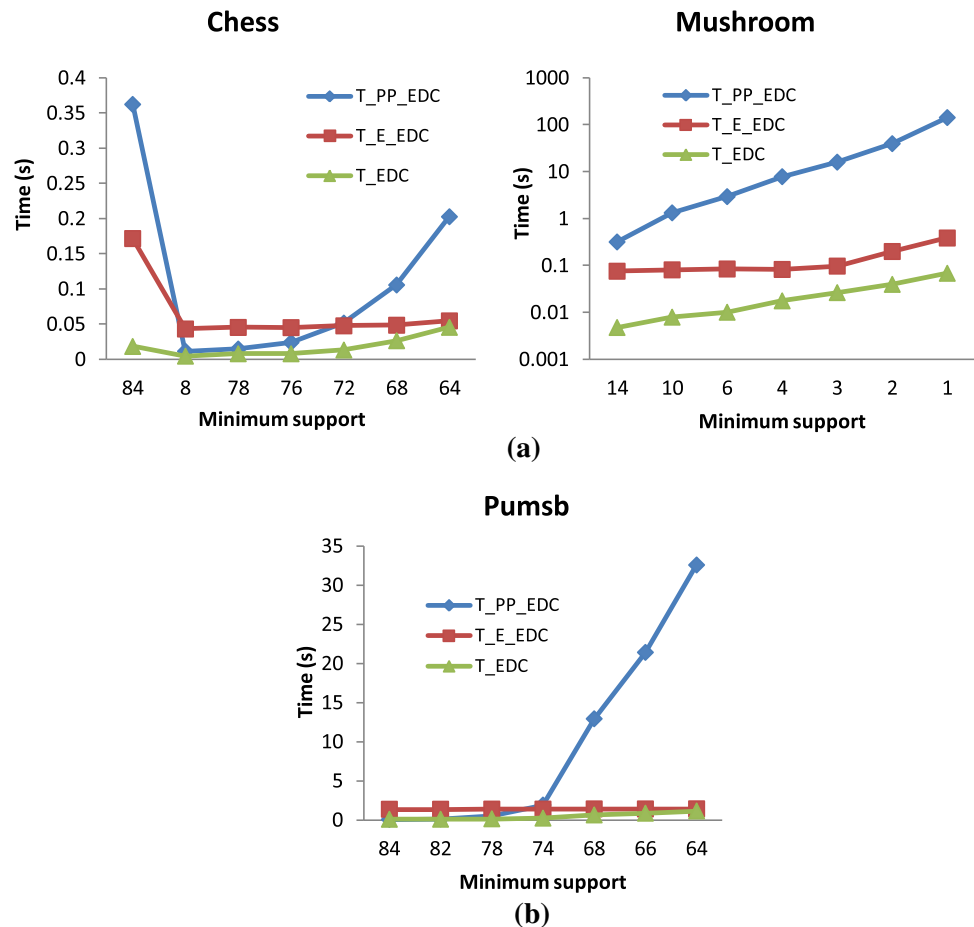


Figure 7a, b show the results in the average execution time of different numbers of constraints. We realize that the performance gap between *MFS-EDC* and *MFS-E-EDC* increases along with the number of constraints (*#Constraints*). The main reason is that, when the extended double constraints changes, *MFS-EDC* executes without creating the lattice of closed itemset and their generators again from the dataset.

In general, *MFS-EDC* outperforms both *MFS-PP-EDC* and *MFS-E-EDC*, especially when the minimum support is lower and the number of constraints is high.

6 Conclusion and future work

In this paper, in theory, checking the general constraints was performed directly on the lattice of closed itemsets and their generators based on partitioning solution set into disjoint equivalence sub-classes. Instead of eliminating an enormous amount of itemsets not satisfying the constraints by so many direct checks, the partition helps to test and eliminate redundant, candidate equivalence classes only based on the necessary conditions. Thereby, the structure and explicit representation of $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$ were shown and

proven to be reliable. On the basis of the theoretical results, on practice, the corresponding algorithm, *MFS-EDC*, to find solution set without generating any redundant candidate was obtained. Its efficiency was verified and compared to several post-processing algorithms on a lot of benchmark datasets in the domain.

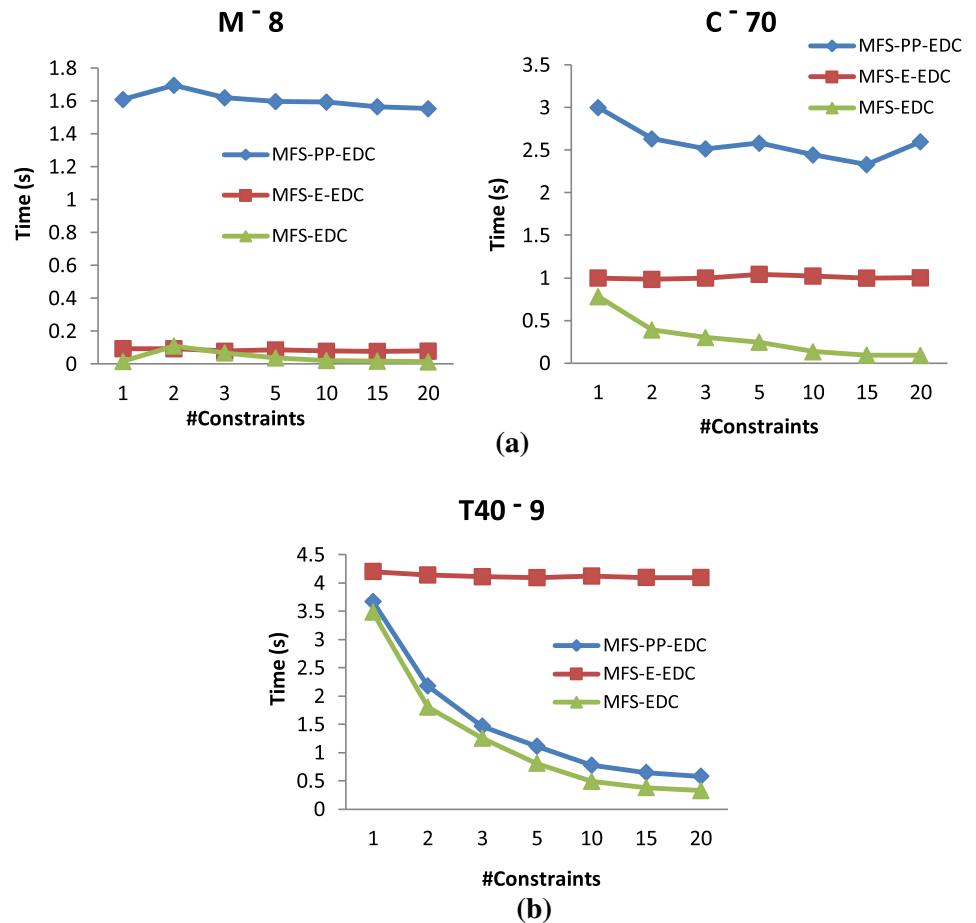
Based on saving a not too large number of the lattice of closed itemsets and their generators, the approach of the paper is sustainable through the regular changes of constraints given by online users. In addition, these generally theoretical results are also the reliable basis for designing parallel algorithms that efficiently mine frequent itemsets with more general constraints in real time.

Appendix

Appendix 1: Proof of Proposition 2

Obviously, $\forall L' \in \mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) \subseteq \mathcal{FS}(s_0, s_1)$, we have $L \stackrel{\text{def}}{=} h(L') \in \mathcal{FCS}(s_0, s_1)$, $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) = \sum_{L \in \mathcal{FCS}(s_0, s_1)} \{[L] \cap \mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)\}$ and $[L] \cap \mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) = \{L' \in [L] | C_{10} \subseteq L' \subseteq$

Fig. 7 **a** Performance results for M-8 (Mushroom, minsup = 8 %) and C-70 (Connect, minsup = 70 %). **b** Performance results for T40-9 (T40I10D100K, minsup = 9 %) for various numbers of constraints



$$L_{C_{11}}, L' \not\subseteq C_{21}\} = \{L' \in \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}} | L' \not\subseteq C_{21}\} = \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}. \quad \square$$

Appendix 2: Proof of Lemma 1 (a)

- (i) We need only prove that $\mathcal{G}(L') \neq \emptyset \Rightarrow L' \neq \emptyset$, the remain of the assertion could see in [4]. Indeed, if $\mathcal{G}(L') \neq \emptyset$, then $\exists L_i \in \mathcal{G}(L') : \emptyset \subset L_i \subseteq L'$, i.e. $L' \neq \emptyset$.
- (ii) “ \subseteq ”: $\forall L_k \in \mathcal{G}_B(L) : L_k \in \mathcal{G}(L), L_k \subseteq B$, we have $L_k \subseteq L_B \subseteq L, L = h(L_k) = h(L_B)$, thus, $L_k \in \mathcal{G}(L_B)$.
 “ \supseteq ”: Conversely, $\forall L_k \in \mathcal{G}(L_B)$, then $L_k \subseteq L_B \subseteq B, h(L_k) = h(L_B)$; since $\mathcal{G}_B(L) \neq \emptyset$, so $\exists L_i \in \mathcal{G}(L) : L_i \subseteq B, L_i \subseteq L_B \subseteq L$ and $L = h(L_i) = h(L_B) = h(L_k)$, i.e. $L_k \in \mathcal{G}(L) : L_k \subseteq B$ or $L_k \in \mathcal{G}_B(L)$.
- (iii) Since $\mathcal{G}_B(L) \neq \emptyset$, so $\exists L_j \in \mathcal{G}(L) : L_j \subseteq L_B \stackrel{\text{def}}{=} L \cap B \subseteq L$, so $h(L_j) = L = h(L_B)$. Denote $L^* \stackrel{\text{def}}{=} A \cup L_j$, then $A \subseteq L^*, \emptyset \subset L_j \subseteq L^* \subseteq L_B \subseteq L, L = h(L_j) = h(L^*) = h(L_B)$ and $L^* \in \mathcal{FS}_{A \subseteq L_B} \neq \emptyset$. Moreover, $\forall L' \in \mathcal{FS}_{A \subseteq L_B}, \forall L_i \in \mathcal{G}(L)$, since $L' \neq \emptyset$, by (i), then for any $L_i \in \mathcal{G}(L') \neq \emptyset$, we

- have $L_i \subseteq L' \subseteq L_B \subseteq B, h(L_i) = h(L') = L$, thus $L_i \in \mathcal{G}_B(L), \mathcal{G}(L') \subseteq \mathcal{G}_B(L)$ and $h(L_i) = L = h(L_B)$.
- (iv) Since $\emptyset \subset U \subseteq V, U$ and V are finite, so there exist $\text{Minimal}(U)$ and $\text{Minimal}(V)$. If $\exists M \in U \cap \text{Minimal}(V)$, and assume that $M \notin \text{Minimal}(U)$, so $\exists K \in \text{Minimal}(U) \subseteq V$ such that $K \subset M$. It contradicts the hypothesis $M \in \text{Minimal}(V)$!

- (b) Consider $\forall L' \subseteq A : L' \neq \emptyset, L' \supseteq A, h(L') = L : \exists L_i \in \mathcal{G}(L'), \text{Cond}(L_i)$. Then, $K_i \stackrel{\text{def}}{=} L_i \setminus A \in U$ and $U \neq \emptyset$. For any $K_k \stackrel{\text{def}}{=} L_k \setminus A \in U : L_k \in \mathcal{G}(L') \subseteq \mathcal{G}(L), \text{Cond}(L_k)$, then $L_k \in \mathcal{G}(L)$ and $K_k \in V$, thus, $U \subseteq V$. Due to $\emptyset \subset U \subseteq V, U$ and V is finite, then $\text{Minimal}(V) \neq \emptyset, \text{Minimal}(U) \neq \emptyset$. Consider $\forall K_j \in \text{Minimal}(U)$, then $L' \supseteq A \cup L_j = A + K_j$ and $K_j \in \text{Minimal}(V)$. Indeed, assume $K_j \notin \text{Minimal}(V)$, then $\exists K_k \stackrel{\text{def}}{=} L_k \setminus A \in \text{Minimal}(V) : K_k \subset K_j, L_k \in \mathcal{G}(L)$ and $\text{Cond}(L_k)$; since $L_k \subseteq A + K_k \subseteq A + K_j \subseteq L' \subseteq L$, so $L = h(L_k) = h(L')$, $L_k \in \mathcal{G}(L'), K_k \in U \cap \text{Minimal}(V)$, thus $K_k \in \text{Minimal}(U)$: it is a contradiction of the facts $K_j \in \text{Minimal}(U)$ and $K_k \subset K_j$! Hence, $\text{Minimal}(U) \subseteq \text{Minimal}(V)$. We can always assume that i is the minimum index in all ones of K_i in $\text{Minimal}(U)$.

(c) If $L' = A + K_i + K'_i + K_i^\sim$, where $K_i \in K_{\min}$, $K'_i \subseteq K_{U,i}$, $K_i^\sim \subseteq K_{-,i}$, then, since $K_i \subseteq L_B$, $K_{-,i} \subseteq L_B$, so $K_{U,i} \subseteq K_{U,i}^{-1} \subseteq L_B$. Hence, $A \subseteq L' \subseteq L_B$. If $K_i \stackrel{\text{def}}{=} L_i \setminus A$, with $L_i \in \mathcal{G}(L)$, then $\emptyset \subset L_i \subseteq A + K_i \subseteq L' \subseteq L_B \subseteq L$, so $L' \neq \emptyset$, $L = h(L') = h(L_i) = h(L_B)$ and $L' \in \mathcal{FS}_{A \subseteq L_B}$.

□

Appendix 3: Proof of Proposition 3

+“(a). \Rightarrow và b. \Leftarrow ”: $\forall L' \in \mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$, $L' \in \mathcal{FS}(s_0, s_1)$, $C_{10} \subseteq L' \subseteq C_{11}$, $L' \not\subseteq C_{21}$, call $L = h(L')$, $L_i \in \mathcal{G}(L') \subseteq \mathcal{G}(L)$ (by Lemma 1a(i)) then $\text{supp}(L) = \text{supp}(L') \in [s_0, s_1]$, $C_{10} \subseteq L' \subseteq L_{C_{11}}$, $\emptyset \subset L_i \subseteq L_i \cup C_{10} \subseteq L' \subseteq L_{C_{11}} \subseteq L$ and $L = h(L_i) = h(L') = h(L_{C_{11}})$. Due to $\emptyset \subset L' \setminus C_{21} = L' \setminus L_{C_{21}} \subseteq L_{C_{11}} \setminus C_{21}$, then $L' \in \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}$, $L \in \mathcal{FCS}(s_0, s_1)$, $C_{10} \subseteq L_{C_{11}} \not\subseteq C_{21}$ and $L_i \in \mathcal{G}_{C_{11}}(L) \neq \emptyset$, i.e. $L \in \mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) \neq \emptyset$.

“(a). \Leftarrow ”: $\forall L \in \mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$, $C_{10} \subseteq L_{C_{11}} \not\subseteq C_{21}$, $\mathcal{G}_{C_{11}}(L) \neq \emptyset$, let $L_i \in \mathcal{G}_{C_{11}}(L) \subseteq \mathcal{G}(L)$: $L_i \subseteq C_{11}$ and $L' \stackrel{\text{def}}{=} (C_{10} \cup L_i) \cup \{a\}$, where $a \in L_{C_{11}} \setminus C_{21} \neq \emptyset$. Then, $a \in L' \setminus C_{21} \neq \emptyset$, so $C_{10} \subseteq L' \subseteq L_{C_{11}} \subseteq C_{11}$, $L' \not\subseteq C_{21}$, $\emptyset \subset L_i \subseteq L' \subseteq L_{C_{11}} \subseteq L$ and $L = h(L_i) = h(L') = h(L_{C_{11}})$, $\text{supp}(L') = \text{supp}(L) \in [s_0, s_1]$, $L' \in \mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) \neq \emptyset$.

+“(b). \supseteq ”: It is a consequence of Proposition 2 and $\mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1) \subseteq \mathcal{FCS}(s_0, s_1)$. □

Appendix 4: Proof of Proposition 4

Note that, $\mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}} = [L]_{C_- \subseteq L_{C^+}, +[L]_{C^*} + L_{C_{10} \subseteq C_{11}, G \not\subseteq C_{21}}}$. Consider $\forall L \in \mathcal{FCS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}(s_0, s_1)$.

(i) + The uniqueness in the representation of $L'' \in \mathcal{FS}_{C_- \subseteq L_{C^+}}$: Assume that L' has two representations: $L'' = C_- + K_k + K'_k + K_k^\sim = C_- + K_i + K'_i + K_i^\sim$, where $1 \leq k < i$, K_k and $K_i \in K_{\min, C_- \subseteq L_{C^+}}$, $K'_k \subseteq K_{U, \min, C_- \subseteq L_{C^+}, i}$, $K_i^\sim \subseteq K_{-, C_- \subseteq L_{C^+}, i}$, $K'_k \subseteq K_{U, C_- \subseteq L_{C^+}, k}$, $K_k^\sim \subseteq K_{-, C_- \subseteq L_{C^+}, k}$. Then, $K_k \subseteq K_i + K'_i + K_i^\sim$, but due to $K_k \subseteq K_{U, C_- \subseteq L_{C^+}} \subseteq K_{U, C_- \subseteq L_{C^+}}^i$, $K_i^\sim \cap K_{U, C_- \subseteq L_{C^+}}^i = \emptyset$, so $K_k \cap K_i^\sim = \emptyset$ and $K_k \subset K_i + K'_i$ (the equality does not happen, because K_k and K_i are two different minimal sets in $K_{\min, C_- \subseteq L_{C^+}}$): it is a contradiction of the selection of K_i !

+ The uniqueness in the representation of $L' \in \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}$: Assume that L' have two representations: $L' = C_{10} + K_k + K'_k + K_k^\sim = C_{10} + K_i + K'_i + K_i^\sim$, where $M \leq k < i$, $K_i \in K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}$, $K_k \in K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^+$, $K_k \neq K_i$,

$K'_i \subseteq K_{U, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}, i}$, $K_i^\sim \subseteq K_{-, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}, i}$, $K'_k \subseteq K_{U, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}, k}$, $K_k^\sim \subseteq K_{-, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}, k}$. Then, $K_k \subseteq K_i + K'_i + K_i^\sim$, but since $K_k \subseteq K_{U, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^k$, $K_i^\sim \cap K_{U, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^k = \emptyset$, so $K_k \cap K_i^\sim = \emptyset$ and $K_k \subset K_i + K'_i$ (the equality does not also happen, because K_k and K_i are two different minimal sets in $K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}$): it contradicts the selection of K_i !

(ii) + “ $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}} \subseteq \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}^*$ ”: We will prove that $[L]_{C_- \subseteq L_{C^+}, +L_{C^*}} \subseteq \mathcal{FS}_{C_- \subseteq L_{C^+}, +L_{C^*}}^*$ and $[L]_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}} \subseteq \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}^*$, thus, $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}} \subseteq \mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}^*$.

• “[L] $_{C_- \subseteq L_{C^+}, +L_{C^*}} \subseteq \mathcal{FS}_{C_- \subseteq L_{C^+}, +L_{C^*}}^*$ ”: $\forall L' \in \mathcal{FS}_{C_{10} \subseteq C_{11}, \not\subseteq C_{21}}$, $\exists L_i \in \mathcal{G}_{C^+}(L) \subseteq \mathcal{G}(L)$: $L_i \subseteq L'$, then $L' \neq \emptyset$ and $L' \supseteq C_{10}$, $L_i \subseteq L' \subseteq L_{C_{11}} \subseteq C_{11}$, $L_i \subseteq C^+ \subseteq C_{21}$. Since $L' \subseteq L_{C_{11}}$, so based on C_{21} , we can partition L' into disjoint subsets $L' \stackrel{\text{def}}{=} L'' + L^\sim$, where $C_- \subseteq L'' \stackrel{\text{def}}{=} L' \cap C_{21} \subseteq L_{C_{11}} \cap C_{21} = L_{C^+} \subseteq L_{C_{21}}$ and $C_{10} \setminus C_{21} \subseteq L^\sim \stackrel{\text{def}}{=} L' \setminus C_{21} \subseteq L_{C_{11}} \setminus C_{21} = L_{C^*}$. Moreover, due to $L'' \setminus C_{21} = \emptyset$, $L^\sim \cap C_{21} = \emptyset$, $L' \not\subseteq L_{C_{21}}$, so $\emptyset \subset L' \setminus L_{C_{21}} = L^\sim \setminus L_{C_{21}} = L^\sim$, $L^\sim \in 2^{[C_{10} \setminus C_{21} \subseteq L_{C^*}]^*}$. We need to prove that $L'' \in \mathcal{FS}_{C_- \subseteq L_{C^+}}^*$. Let $K_i \stackrel{\text{def}}{=} L_i \setminus C_- = L_i \setminus C_{10}$ ($\forall L_i \subseteq C_{21}$), $\text{Cond}(L_k) \stackrel{\text{def}}{=} (L_k \subseteq C^+)$, $U \stackrel{\text{def}}{=} \{K_k \stackrel{\text{def}}{=} L_k \setminus C_- \mid L_k \in \mathcal{G}(L''), \text{Cond}(L_k)\}$, $V \stackrel{\text{def}}{=} \{K_k \stackrel{\text{def}}{=} L_k \setminus C_- \mid L_k \in \mathcal{G}(L), \text{Cond}(L_k)\} = \{K_k \stackrel{\text{def}}{=} L_k \setminus C_- \mid L_k \in \mathcal{G}_{C^+}(L)\}$, then $L_i \subseteq C_- + K_i = C_- \cup L_i \subseteq L'' \subseteq L$, $h(L_i) = h(L'') = L$ and $L_i \in \mathcal{G}(L'')$, thus, $K_i \in U \subseteq V$, $K_{\min, C_- \subseteq L_{C^+}} = \text{Minimal}(V)$. From Lemma 1b, we can always assume that i is the minimum index such that $K_i \in \text{Minimal}(U)$. Then, $K_i \in \text{Minimal}(V)$ và $L'' \supseteq C_- + K_i$. We can represent L'' as follows: $L'' \stackrel{\text{def}}{=} C_- + K_i + K'_i + K_i^\sim$, where $K'_i \stackrel{\text{def}}{=} [L'' \setminus (C_- + K_i)] \cap K_{U, C_- \subseteq L_{C^+}}^i \subseteq K_{U, C_- \subseteq L_{C^+}}^i \setminus K_i = K_{U, C_- \subseteq L_{C^+}, i}$, $K_i^\sim \stackrel{\text{def}}{=} [L'' \setminus (C_- + K_i)] \setminus K_{U, C_- \subseteq L_{C^+}}^i \subseteq K_{-, C_- \subseteq L_{C^+}, i}$. Finally, assume that the condition (*) is false, i.e. $\exists K_k \equiv L_k \setminus C_- \in K_{\min, C_- \subseteq L_{C^+}}$: $1 \leq k < i$ and $K_k \subset K_i + K'_i$, then $L_k \in \mathcal{G}_{C^+}(L) \subseteq \mathcal{G}(L)$, $L_k \subseteq C_- + K_k \subseteq C_- + K_i + K'_i \subseteq L'' \subseteq L$ and $L_k \in \mathcal{G}(L'')$. Hence, $K_k \in U \cap \text{Minimal}(V)$, $K_k \in \text{Minimal}(U)$ and $k < i$: it contradicts the selection of the index i ! Thus, $L'' \in \mathcal{FS}_{C_- \subseteq L_{C^+}}^*$.

• “[L] $_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}} \subseteq \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^*$ ”: cxv $\forall L' \in [L]_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}$, we have $L' \in \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}$, $\exists L_i \in \mathcal{G}_{C_{11}, \not\subseteq C_{21}}(L) \subseteq \mathcal{G}(L)$: $L_i \subseteq L'$, so $L' \neq \emptyset$, $L' \supseteq C_{10}$, $L_i \subseteq L' \subseteq L_{C_{11}} \subseteq C_{11}$, $L_i \not\subseteq C_{21}$ and $(L_k \not\subseteq L', \forall L_k \in \mathcal{G}_{C^+}(L))^{(-)}$. From Lemma 1a, we have $h(L') = L$ and $L_i \in \mathcal{G}(L')$. Let $\text{Cond}'(L_k) \stackrel{\text{def}}{=} (L_k \not\subseteq C_{21}, L_k \subseteq C_{11})$, $U \stackrel{\text{def}}{=} \{K_i \stackrel{\text{def}}{=} L_i \setminus C_{10} \mid L_k \in \mathcal{G}(L'), \text{Cond}'(L_k)\}$,

$V \stackrel{\text{def}}{=} \{K_i \stackrel{\text{def}}{=} L_i \setminus C_{10} | L_k \in \mathcal{G}(L), \text{Cond}'(L_k)\} = \{K_i \stackrel{\text{def}}{=} L_i \setminus C_{10} | L_k \in \mathcal{G}_{C_{11}, \not\subseteq C_{21}}(L)\}$. From Lemma 1b, due to $K_i \stackrel{\text{def}}{=} L_i \setminus C_{10} \in U$, then $\emptyset \neq U \subseteq V$ and we can choose the minimum index i such that $K_i \in \text{Minimal}(U)$. Then, $L' \supseteq C_{10} + K_i$ and $K_i \in \text{Minimal}(V) \equiv K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}} \neq \emptyset$. Since $C_{10} + K_i \subseteq L' \subseteq L_{C_{11}}$, call $L_i'' \stackrel{\text{def}}{=} L' \setminus (C_{10} + K_i) \subseteq L_{C_{11}} \setminus (C_{10} + K_i)$, so $L' = (C_{10} + K_i) + L_i'' = C_{10} + K_i + K_i' + K_i^\sim$, where $K_i' = L_i'' \cap K_{U, C_{10} \subseteq L_{C_{11}}}^{i-1} \subseteq K_{U, C_{10} \subseteq L_{C_{11}}}^{i-1} \setminus K_i = K_{U, C_{10} \subseteq L_{C_{11}}, i}$, $K_i^\sim = L_i'' \setminus K_{U, C_{10} \subseteq L_{C_{11}}}^{i-1} \subseteq L_{C_{11}} \setminus (C_{10} + K_i + K_{U, C_{10} \subseteq L_{C_{11}}, i}) = K_{-, C_{10} \subseteq L_{C_{11}}, i}$. Finally, assume that the condition $(^{**})$ is not true, i.e. $\exists K_i \stackrel{\text{def}}{=} L_i \setminus C_{10} \in K_{\min, C_{10} \subseteq L_{C_{11}}, C_{21}^+}$: $1 \leq k < i$ and $K_k \subseteq K_i + K_i'$, then $L_k \in \mathcal{G}_{C_{11}}(L) \subseteq \mathcal{G}(L)$, $L_k \subseteq C_{11}$, $L_k \subseteq C_{10} + K_k \subseteq C_{10} + K_i + K_i' \subseteq L' \subseteq L$ and $h(L_k) = h(L') = L$. Hence, $L_k \in (L')$, $L_k \not\subseteq C_{21}$ (because if $L_k \subseteq C_{21}$, then $L_k \subseteq C^+$ and $L_k \in \mathcal{G}_{C^+}(L)$: it contradicts the above condition $(^{-})$!). Moreover, $L_k \in \mathcal{G}_{C_{11}, \not\subseteq C_{21}}(L)$, $K_k \in K_{\min, C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21} \cap U}$, $K_k \in \text{Minimal}(U)$ and $k < i$: it is a contradiction of the selection the minimum index i such that $K_i \in \text{Minimal}(U)$! Thus, the condition $(^{**})$ in $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^*$ is true and $L' \in \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^*$.
 + “ $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}^* \subseteq \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}$ ” We will prove that $\mathcal{FS}_{C_- \subseteq L_{C^+}, +L_{C^*}}^* \subseteq [L]_{C_- \subseteq L_{C^+}, +L_{C^*}}$ and $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^* \subseteq [L]_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}$, thus, $\mathcal{FS}_{C_- \subseteq L_{C^+}, +L_{C^*}}^* \cap \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^* = \emptyset$ and $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}^* \subseteq \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}$.
 . “ $\mathcal{FS}_{C_- \subseteq L_{C^+}, +L_{C^*}}^* \subseteq [L]_{C_- \subseteq L_{C^+}, +L_{C^*}}$ ”: $\forall L' \stackrel{\text{def}}{=} L'' + L^\sim \in \mathcal{FS}_{C_- \subseteq L_{C^+}, +L_{C^*}}^*$, $L'' \stackrel{\text{def}}{=} C_- + K_i + K_i' + K_i^\sim \in \mathcal{FS}_{C_- \subseteq L_{C^+}}^*$, $L^\sim \in 2^{[C_{10} \setminus \not\subseteq C_{21} \subseteq L_{C^*}]}$, where $K_i \stackrel{\text{def}}{=} L_i \setminus C_- = L_i \setminus C_{10}$ (because $L_i \in \mathcal{G}_{C^+}(L) \subseteq \mathcal{G}(L)$, $L_i \subseteq C^+ \not\subseteq C_{21}$) and from Lemma 1c, $L'' \in \mathcal{FS}_{C_- \subseteq L_{C^+}}$. On the other hand, since $C_{10} = (\neq) + (C_{10} \not\subseteq C_{21}) \subseteq L' = L^\sim + L'' \subseteq L_{C^*} + L_{C^+} = L_{C_{11}}$, $L_i \subseteq C_- + K_i \subseteq L'' \subseteq L' \subseteq L_{C_{11}} \subseteq L$, $h(L_{C_{11}}) = h(L') = h(L_i) = L$, $\emptyset \subset L^\sim = L' \setminus C_{21} \subseteq L' \setminus C_{21}$, then $L' \not\subseteq C_{21}$. Hence, $L' \in \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}$ and $L' \in \mathcal{FS}_{C_- \subseteq L_{C^+}, +L_{C^*}}$.
 . “ $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^* \subseteq [L]_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}$ ”: $\forall L' \stackrel{\text{def}}{=} C_{10} + K_i + K_i' + K_i^\sim \in \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}^*$, $K_i \stackrel{\text{def}}{=} L_i \setminus C_{10}$, $L_i \in \mathcal{G}_{C_{11}, \not\subseteq C_{21}}(L)$ and $(K_k \not\subseteq K_i + K_i', \forall K_k \in K_{\min, C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}^+ : 1 \leq k < i)^{(**)}$, then from Lemma 1a and c, due to $\mathcal{G}_{C_{11}, \not\subseteq C_{21}}(L) \subseteq \mathcal{G}(L)$, we obtain $h(L') = L$, $L' \in \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}}$ and $L' \setminus C_{21} \supseteq L_i \setminus C_{21} \supset \emptyset$, $L_i \subseteq K_i +$

$C_{10} \subseteq L'$, thus $L' \in \mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}$. To prove $L' \in [L]_{C_{10} \subseteq L_{C_{11}}, G \not\subseteq C_{21}}$, in addition, we need to test the condition $(^{-})$: $(L_k \not\subseteq L', \forall L_k \in \mathcal{G}_{C^+}(L))$ by contradiction. Assume that $\exists L_k \in \mathcal{G}_{C^+}(L) \subseteq \mathcal{G}_{C_{11}}(L) \subseteq \mathcal{G}(L)$: $L_k \subseteq L' = C_{10} + K_i + K_i' + K_i^\sim \subseteq L$, $L_k \subseteq L_{C^+} \not\subseteq C_{21}$, then $L_k \in \mathcal{G}(L')$ and $k \leq M < i$ (from the above indexing k of K_k). Using Lemma 1b, with $U'' \stackrel{\text{def}}{=} \{K_i \stackrel{\text{def}}{=} L_i \setminus C_{10}, L_i \in \mathcal{G}(L'), \text{Cond}(L_k)\}$ and $K_i \stackrel{\text{def}}{=} L_i \setminus C_{10} = L_k \setminus C_- \in U'' \neq \emptyset$, $U'' \subseteq V$, we can choose the minimum index k such that $K_k \in \text{Minimal}(U'')$ and $K_k \in \text{Minimal}(V) = K_{\min, C_- \subseteq L_{C^+}} \subseteq K_{\min, C_{10} \subseteq L_{C_{11}}, C_{21}^+}$. Since $K_k \subseteq L' \setminus C_{10} = K_i + K_i' + K_i^\sim$, $K_k \subseteq K_{U, C_- \subseteq L_{C^+}}^k = K_{U, C_{10} \subseteq L_{C_{11}}}^k \subseteq K_{U, C_{10} \subseteq L_{C_{11}}}$ and $K_i^\sim \cap K_{U, C_{10} \subseteq L_{C_{11}}}^k = \emptyset$, so $K_k \cap K_i^\sim = \emptyset$ and $K_k \subseteq K_i + K_i'$: It is a contradiction of the selection of the index i in $(^{**})$! Thus, the condition $(^{-})$ is true.

Finally, all itemsets in $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}$ have a unique representation and are generated completely and distinctly by itemsets in $\mathcal{FS}_{C_{10} \subseteq L_{C_{11}}, \not\subseteq C_{21}}^*$. \square

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Agrawal, R., Imielinski, T., Swami, N: Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD, pp. 207–216 (1993)
2. Anh, T., Hai, D., Tin, T., Bac, L.: Efficient algorithms for mining frequent Itemsets with constraint. In: Proceedings of the Third International Conference on Knowledge and Systems Engineering, pp. 19–25 (2011)
3. Anh, T., Hai, D., Tin, T., Bac, L.: Mining frequent itemsets with dualistic constraints. In: Proceedings PRICAI 2012, LNAI, vol. 7458, pp. 807–813 (2012)
4. Anh, T., Tin, T., Bac, L.: Simultaneous mining of frequent closed itemsets and their generators: foundation and algorithm. Int. J. Eng. Appl. Artif. Intell. (EAAI) **36**, 64–80 (2014)
5. Bayardo, R.J., Agrawal, R., Gunopulos, D.: Constraint-based rule mining in large, dense databases. Proc. Data Min. Knowl. Discov. **4**, 217–240 (2000)
6. Bonchi, F., Giannotti, F., Mazzanti, A., Pedreschi, D.: Examiner: optimized level-wise frequent pattern mining with monotone constraints. In: Proceedings IEEE ICDM'03, pp. 11–18 (2003)
7. Bonchi, F., Lucchese, C.: On closed constrained frequent pattern mining. In: Proceedings IEEE ICDM'04, pp. 35–42 (2004)
8. Boulicaut, J.F., Bykowski, A., Rigotti, C.: Free-sets: a condensed representation of boolean data for the approximation of frequency queries. Data Min. Knowl. Dis. **7**, 5–22 (2003)
9. Boulicaut, J.F., Jedy, B.: Using constraints during set mining: should we prune or not. In: Actes des Seizime Jounes Bases de Données Avances BDA'00, Blois, pp. 221–237 (2000)

10. Bucila, C., Gehrke, J.E., Kifer, D., White, W.: Dualminer: a dual-pruning algorithm for itemsets with constraints. *Data Min. Knowl. Dis.* **7**, 241–272 (2003)
11. Burdick, D., Calimlim, M., Gehrke, J.: MAFIA: A maximal frequent itemset algorithm for transactional databases. In: *Proceedings IEEE ICDE'01*, pp. 443–452 (2001)
12. Dong, J., Han, M.: BitTable-FI: an efficient mining frequent itemsets algorithm. *Int. J. Knowl. Based Syst.* **20**, 329–335 (2007)
13. Frequent Itemset Mining Dataset Repository (FIMDR). <http://fimi.cs.helsinki.fi/data/>. Accessed 2009
14. Grahne, G., Zhu, J.: Fast algorithms for frequent itemset mining using fp-trees. *Proc. IEEE Trans. Knowl. Data Eng.* **17**, 1347–1362 (2005)
15. Hai, D., Tin, T.: An efficient method for mining association rules based on minimum single constraints. *Vietnam J. Comput. Sci.* **2**, 67–83 (2015)
16. Hai, D., Tin, T., Bac, L.: An efficient algorithm for mining frequent itemsets with single constraint. *Adv. Comput. Methods Knowl. Eng. Sci.* **479**, 367–378 (2013)
17. Hai, D., Tin, T., Bay, V.: An efficient method for mining frequent itemsets with double constraints. *Int. J. Eng. Appl. Artif. Intell. (EAAI)* **27**, 148–154 (2014)
18. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: *SIGMOD'00*, pp. 1–12 (2000)
19. Huy, P., Tin, T.: An efficient lattice-based approach for generator mining. *Int. J. Adv. Comput. Res.* **4**, 741–751 (2014)
20. Jeudy, B., Boulicaut, J.F.: Optimization of association rule mining queries. *Intell. Data Anal.* **6**, 341–357 (2002)
21. Lin, D.I., Kedem, Z.M.: Pincer search: an efficient algorithm for discovering the maximum frequent sets. *IEEE Trans Knowl. Data Eng.* **14**, 553–566 (2002)
22. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. *Data Min. Knowl. Dis.* **1**, 241–258 (1997)
23. Mashoria, V., Singh, A.: A survey of mining association rules using constraints. *Int. J. Comput. Technol.* **7**, 620–625 (2013)
24. Nguyen, R.T., Lakshmanan, V.S., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained association rules. In: *Proceedings of the 1998 ACM-SIG-MOD International Conference on the Management of Data*, pp. 13–24 (1998)
25. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Efficient mining of association rules using closed itemset lattices. *Inf. Syst.* **24**(1), 25–46 (1999)
26. Pasquier, N., Taouil, R., Bastide, Y., Stumme, G., Lakhal, L.: Generating a condensed representation for association rules. *Intell. Inf. Syst.* **24**, 29–60 (2005)
27. Pei, J., Han, J.: Constrained frequent pattern mining: a pattern-growth view. *Proc. ACM SIGKDD Explor.* **4**, 31–39 (2002)
28. Pei, J., Han, J., Lakshmanan, L.V.S.: Mining frequent itemsets with convertible constraints. In: *Proceedings IEEE ICDE'01*, pp. 433–442 (2001)
29. Song, W., Yang, B., Xu, Z.: Index-BitTableFI: an improved algorithm for mining frequent itemsets. *Int. J. Knowl. Based Syst.* **21**, 507–513 (2008)
30. Szathmary, L., Valtchev, P., Napoli, A.: Efficient vertical mining of frequent closed itemsets and generators. *IDA* **2009**, 393–404 (2013)
31. Tin, T., Anh, T.: Structure of set of association rules based on concept lattice. In: *Advances in Intelligent Information and Database Systems, SCI*, vol. 283, pp. 217–227. Springer (2010)
32. Vo, B., Hong, T.P., Le, B.: A lattice-based approach for mining most generalization association rules. *Knowl. Based Syst.* **45**, 20–30 (2013)
33. Vo, B., Hong, T.P., Le, B.: DBV-Miner: a dynamic bit-vector approach for fast mining frequent closed itemsets. *Expert Syst. Appl.* **39**, 7196–7206 (2012)
34. Zaki, M.J.: Mining non-redundant association rules. *Data Min. Knowl. Discov.* **9**(3), 223–248 (2004)
35. Zaki, M.J., Hsiao, C.J.: Efficient algorithms for mining closed itemsets and their lattice structure. *IEEE Trans. Knowl. Data Eng.* **17**, 462–478 (2005)
36. Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of association rules. In: *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD'97)*, pp. 283–296 (1997)